NONLINEAR ANALYSIS OF SPEECH FROM A SYNTHESIS PERSPECTIVE

Michael Banbrook

A thesis submitted for the degree of Doctor of Philosophy.

The University of Edinburgh

October 15, 1996

Abstract

With the emergence of nonlinear dynamical systems analysis over recent years it has become clear that conventional time domain and frequency domain approaches to speech synthesis may be far from optimal. Using state space reconstructions of the time domain speech signal it is, at least in theory, possible to investigate a number of invariant geometrical measures for the underlying system which give a more thorough understanding of the dynamics of the system and therefore the form that any model should take. This thesis introduces a number of nonlinear dynamical analysis tools which are then applied to a database of vowels to extract the underlying invariant geometrical properties. The results of this analysis are then applied, using ideas taken from nonlinear dynamics, to the problem of speech synthesis and a novel synthesis technique is described and demonstrated.

The tools used for the analysis are time delay embedding, singular value decomposition, correlation dimension, local singular value analysis, Lyapunov spectra and short term prediction properties. Although there have been many papers written about these tools, and algorithms proposed, there are currently no generally accepted techniques, especially for the calculation of Lyapunov spectra in the presence of noise and data length limitations. This thesis introduces all of the above tools and looks in detail at Lyapunov exponents and two major novel modifications are proposed that are demonstrated to be more robust than conventional techniques.

The novel robust techniques are applied to a large database of vowel sounds showing that the vowels tested show evidence of nonlinear, low-dimensional, non-chaotic behaviour. It is particularly the evidence of non-chaotic behaviour that is of importance from a synthesis point of view and is used in the final section of the thesis which introduces a novel synthesis technique. The synthesis technique, which is based on ideas taken from nonlinear dynamics theory is detailed and demonstrated showing that it is capable of high quality natural sounding speech.

Declaration of originality

I hereby declare that this thesis and the work reported herein was composed and originated by myself, in the Department of Electrical Engineering at the University of Edinburgh.

Michael Banbrook

Acknowledgements

I would like to thank the following people for their invaluable assistance during the course of this PhD:

- My wife, Andrea.
- Steve McLaughlin, my University supervisor, for his support and guidance.
- Andrew Lowry and the Speech Synthesis group of BT Labs, Martlesham for both general and financial support.
- Gary Ushaw for generally keeping me sane.

Contents

List of Figures

List of Tables

Abbreviations

GUI	Graphical user interface
LPC	Linear predicitive coding
PCA	Principle component analysis
PSOLA	Pitch synchronous overlap and add
RBF	Radial basis function
SAMPA	Speech assessment methodology phonetic alphabet
SVD	Singular value decomposition

List of principal symbols

Fn	Frequency of the nth formant
$\underline{s}$	Vector in state space
$f(.)$	Mapping function
ω	Frequency of oscillation
D_i	Dimension of an intersection manifold
$x(t)$	Time series
m	Embedding dimension
w	SVD window length
$\underline{x}_i$	Element of embedded time series
d_{cap}	Capacity dimension
d_i	Information dimension
P_i	Probability of a point being in the ith box
$C(.)$	Correlation function
$\delta()$	Heaviside step function
$\|\|\ \|\|$	Euclidean norm
d_L	Lyapunov dimension
λ_i	Ordered Lyapunov exponent
$eig(.)$	Eigenvalue
$J(.)$	Jacobian
$\boldsymbol{S}$	Matrix of singular vectors
$\boldsymbol{C}$	Matrix of singular vectors
Σ	Matrix of singular values
Θ	Structure matrix
Ξ	Covariance matrix
$\hat{\boldsymbol{X}}$	SVD reduced embedded time series
$\boldsymbol{T}$	Trajectory matrix

$\boldsymbol{B}$	Neighbourhood matrix
$\boldsymbol{\Gamma}_i$	Neighbourhood set for $\underline{x}_i$
$\lvert . \rvert$	Modulus
γ_n	nth row of $\hat{\boldsymbol{X}}$ indicated by $\boldsymbol{\Gamma}_i$
$\boldsymbol{I}$	Identity matrix
$\underline{p}$	Predicted point in state space
$E_m(.)$	Normalised geometric mean error
σ	Standard deviation
τ_d	Time delay for time delay embedding (in samples)
k	Number of nearest neighbours used to construct local model
$\underline{t}_i$	Point on an intermediate attractor
$\underline{e}_i$	Starting attractor for morphing operation
$\underline{a}_i$	End attractor for morphing operation
$\underline{s}_i$	Synthesised vector
$(.)^+$	Pseudo inverse operation
$(.)^T$	Transposition operation
$(.)^{-1}$	Inverse operation

Chapter 1

INTRODUCTION

The ability to speak is a gift that many of us take for granted. For most of us we can not remember our early formative years during which we gurgled, cried and tried in vain to make the sounds that adults make as a matter of course. Consequently it is of no surprise that when it comes to synthesis of speech we naively believe that the problem is easily solvable, especially with the phenomenal power of modern computers to simultaneously play the role of our brain and provide a model of our physiological speech production apparatus. Unfortunately life is seldom as simple as we believe and even though we can now produce very high quality synthesised voices there is always some elusive quality that is missing. It is the search for this mystery ingredient that has been the driving force behind many recent applications of new and emergent theories to the problem of speech, in particular the 'new science' of nonlinear dynamics and chaos theory has received a great deal of attention. Unfortunately, as is often the case with emergent theories, many of the analysis techniques and possible applications are tried and tested in clean, artificial conditions and therefore do not perform quite as expected when applied to 'real world' signals which contain background noise and are not stationary.

The aim of this thesis is to provide an analysis of vowel sounds, taking into account the problems of data length and noise contamination, and show how these results may be applied to speech synthesis through a novel technique which uses the state space domain, rather than the conventional time or frequency domains, to allow the dynamics of the underlying system to be modelled. The contribution of this work is threefold: firstly the work improves on current chaotic systems analysis to allow for short and noisy signals; secondly a comprehensive analysis of vowels is carried out showing that vowels exhibit low dimensional, nonlinear, non-chaotic behaviour; and thirdly a novel synthesis technique is described and demonstrated. The purpose of this chapter is to introduce this thesis and show the motivation behind, and the contribution of, the work presented herein. The chapter has a discussion of the current state of speech synthesis and work being conducted both on chaotic analysis in general and on speech itself. Finally the organisation of the thesis is described.

1.1 Motivation

One of mankind's greatest dreams is to be able to synthesise speech to such an extent that it is imperceptible from real speech. This dream has often had to be sacrificed because unless the prosodic models used to generate the speech are perfect then the result can be that the speech becomes less intelligible. Currently it is certainly the case that the modifications that are made to speech synthesisers to produce more natural speech have traditionally had the side effect of lowering the intelligibility. This would suggest that the modifications are in fact not quite recreating all the important features of true speech; something is still missing. Just exactly what the missing feature actually is does not seem to be clear with many different techniques being postulated to overcome it:

- synthesisers have been advanced to include prosodic information, such as intonation and stress [3–6], and models of jitter and shimmer [7–12],
- inclusion of nonlinearities [13],
- complicated glottal flow models [14],
- accurate physical models such as multiple mass models of the vocal folds [15],
- new theories on sound propagation through the vocal tract including turbulence and vortex shedding [16,17].

A common thread in many of these ideas is that the traditional synthesis models are based on linear systems even though there is plenty of evidence to suggest that speech is in fact a nonlinear process. Clearly if this is the case then the missing information could be the result of approximations made by the linear models. This idea does certainly seem to hold water with many of the complicated nonlinear models of glottal flow showing clear improvement over their simpler linear analogues [14]. However even the inclusion of these nonlinear source models does not produce completely natural speech. Some very recent research has suggested that the use of nonlinear models such as neural networks [18,19] or radial basis functions [20] may be the way forward but neither of these seem to have given conclusive results.

Given that speech may be the result of a nonlinear process with a degree of inherent feedback [17] then it is not unreasonable to postulate that the missing information

may be the result of speech being chaotic. Chaos theory is currently a very active field that is being widely applied to a range of disciplines but it is worth stressing, as will be repeatedly done throughout this thesis, that chaos theory is an emergent theory that is still very new and is not easily applicable to signals that originate from 'real world' systems. A range of chaotic analysis tools have been applied to speech with a varied amount of success [18,21–28] as is described in detail in Chapter 2. The only real conclusions that can be drawn from the results are that there is evidence of low dimensional behaviour in vowels and high dimensional behaviour in fricatives, although exact figures do differ from paper to paper. It is this low dimensional behaviour that is the motivation behind the work in this thesis; if vowels are low dimensional then it should be possible to investigate their invariant features and in some way exploit them.

Unfortunately the current state of the art algorithms used for the analysis of chaos are extremely difficult to use and can be easily misapplied. At the onset of this work very few 'canned' routines existed with the exception of the GRASS suite of programs for correlation dimension calculation. Consequently it has been necessary to start from scratch and create a new suite of programs, see Appendix A, capable of supporting a complete analysis of the chaotic properties of a time domain signal. As will become apparent from the chapters on chaos this is no simple problem especially the inclusion of noise robustness. Particularly difficult is the calculation of Lyapunov spectra and over the last few years several algorithms have been suggested although none have reported sufficiently good noise robustness to compete with the algorithm presented in this thesis which involves novel modifications of the technique proposed by Darbyshire and Broomhead [29]. By producing new algorithms which can cope with realistic noise levels it is shown conclusively whether vowels are chaotic or not.

Of course in order for this work to be of any real use then it needs to have an application. This thesis shows how the ideas of nonlinear dynamics and chaos may be applied to speech synthesis giving a complete description and demonstration of a possible novel design for a speech synthesiser.

1.2 Thesis organisation

The Thesis is structured as outlined below. This chapter has introduced the key motivation behind the work. The next two chapters give overviews of the main

theories and current state of the art technology being applied to these fields.

Chapter 2 looks at speech from the perspective of synthesis. The chapter is intended to give the reader a feel for the work that has already been performed in the field and provide a clear indication of the current shortcomings of modern speech synthesis. The chapter is split into the following sections: a brief history of speech analysis and synthesis; details of the phonetics and the physiological background to speech production; common time and frequency domain representations of speech; an overview of current synthesis techniques; and a review of some of the evidence for nonlinear and chaotic behaviour in speech.

Chapter 3 looks at the emergent theories of nonlinear dynamics and chaos. The underlying principles and philosophies are described and a number of analysis tools and techniques are introduced. The primary aim of the chapter is to show the application of chaos theory to clean, artificial systems and provides a short cautionary note regarding the problems of analysing 'real world' data which is dealt with in more detail in the next chapter.

Chapter 4 presents the analysis tools that are used later to investigate the behaviour of vowels, and details the novel improvements that make the tools applicable to real world signals. The tools used are time delay embedding, singular value decomposition, correlation dimension, local singular value analysis, Lyapunov spectra and short term prediction properties. These tools are tested for their abilities to provide accurate results even for noisy or short data sets. It is shown in detail that conventional Lyapunov spectra algorithms can give extremely misleading results for noisy data and novel modifications to the Lyapunov exponent algorithm are described. These new techniques allow the tools to be used with confidence and are thus applied to speech.

In Chapter 5 a range of vowels are analysed using the tools described in chapter 4. The chapter details the data set and the collection technique utilised, followed by a number of sections that explore some relevant invariant geometrical properties: the embedding of the system into state space; the short term predictive properties, showing how these relate to the dimension of the attractor; the local singular value decomposition; and the calculation of the Lyapunov spectra using the novel algorithm developed in Chapter 4. Finally some conclusions are drawn regarding the implications of this work.

Chapter 6 builds on the results from the previous chapter and utilises techniques derived from chaos theory to present a novel synthesis technique. The synthesis is

achieved by implementing the synthesiser entirely in state space allowing the underlying dynamics of the speech to be modelled. The chapter describes the basic elements of a complete synthesiser which is demonstrated, through a limited set of words, to produce high quality speech including the production of both realistic vowels and simple coarticulation effects.

Finally in Chapter 7 the conclusions of this thesis are presented and some ideas for potential areas for further work are suggested.

Chapter 2
SPEECH BACKGROUND

This chapter presents an overview of the techniques used in speech both from an analysis and synthesis perspective. The chapter is intended to give the reader a feel for the work that has already been performed in the field and provide a clear indication of the current shortcomings of modern speech synthesis. The chapter is split into the following sections: a brief history of speech analysis and synthesis; details of the phonetics and the physiological background to speech production; time and frequency domain representations of speech; an overview of current synthesis techniques; and a review of some of the evidence for nonlinear and chaotic behaviour in speech.

2.1 Introduction

Today in the world of high power, high speed computing we seem to take it for granted that speech can be synthesised and that the processes that govern speech production have been researched to their limits and are fully understood. This unfortunately is not the case. Although modern techniques can produce recognisable speech, and recognisers can achieve reasonable error rates, there are whole areas of problems that continue to elude solution. Indeed it is a classic example of a field where the more we find out, the more we realise that we don't actually know.

The development of speech synthesis is recorded in many texts, such as Paget [30], Breen [1], Linggard [31] and Holmes [32]. The rest of this subsection is a brief overview of this material.

Development of interest in the human speech production mechanism first really reared its head around the end of the eighteenth century in the form of the Von Kempelen speaking machine which is a mechanical analogue of the human vocal system. Though the machine is reported to have been able to reproduce speech it required an extremely talented operator who played it in much the same way as a fine musician may play an instrument. This stimulated research into exactly what it is that makes sounds recognisable as speech: Bishop Wilkins classified speech sounds by articulation; Kratzenstien

used resonant cavities to create vowels; Robert Willis attempted to show that vowel characteristics were governed by a single resonance, an idea which was later refuted by Wheatstone and expanded on by Helmholtz, Bell and Lloyd. All this led Miller to synthesise vowels using organ pipes tuned to represent the Fourier components. At around the same time, 1930's, Sir Richard Paget performed numerous experiments, using his own highly 'tuned' hearing, to analyse resonant frequencies and produced a form of early synthesiser which required seven operators orchestrated to produce a full sentence.

Paget's incredible talent for hearing the resonances in speech was overcome by the advent of electronic instrumentation such as the oscilloscope and the spectrogram. The new knowledge of electrical circuits enabled a whole new breed of synthesisers to be produced and it was Dudley that demonstrated the Vocoder to the New York World Fair in 1939. Again, as with previous synthesisers, the machine was 'driven' by skilled operators to produce the required words. Although some attempts to remove the need for the operator were made it was only really the development of the computer that allowed a truly operator-free system to be achieved. By the 1970's the vocoder had been superseded by the ideas of Linear Predictive Coding (LPC) and a whole host of new studies into synthesis models: Fant showed that there are an infinite number of resonances; Flanagan, Ishizaka and Shipley found that frictatives were created by turbulence; resonant circuit models were created by Flanagan and Holmes, Mattingly, and Shearme, whilst transmission lines were used by Dunn and Stevens, Kasowski and Fant. As digital computing advanced so did the possibility for full text to speech systems as demonstrated by the Klatt's synthesiser [4] which set the early standards for all to follow.

Since the Klatt synthesiser there have been many wide ranging improvements and new techniques postulated, as described later in this chapter, but there are still large unsolved problems and controversial new theories emerging.

2.2 Production

Before considering how speech is produced or measured, it is useful to introduce some of the common notation used for defining individual sounds. Speech can be broken down into small segments each of which is unambiguously distinguishable, these segments are represented by the symbolic abstractions called *phonemes*. The

phonemes can be represented by any of a number of different phonetic alphabets, such as SAMPA (speech assessment methodology phonetic alphabet) [1], as in Table 2.1, which allow phoneticians to unambiguously transcribe phrases or words.

p	pet	t	tie	k	key	b	bag	d	dog	g	guy
m	man	n	name	N	sing	f	five	v	via	T	thigh
D	they	s	sigh	z	zoo	S	shy	Z	lounge	l	lip
w	will	r	ring	j	you	h	hill	tS	church	dZ	jive
i	bead	I	bit	E	bed	{	bad	A	card	Q	cod
O	court	U	good	u	food	V	bud	3	bird	@	ago
eI	may	@U	mow	aI	high	OI	boy	I@	heard	e@	rare
U@	Ruhr	aU	cow								

Table 2.1: *SAMPA phonetic symbols*

The mechanism by which we produce speech is a complex interaction of several processes. The basic mechanism for producing any sound is to expel air from the lungs, using muscular action. The expelled air is forced through the larynx, shown in Figure 2.1, which is capable of producing two main categories of speech: voiced and unvoiced.

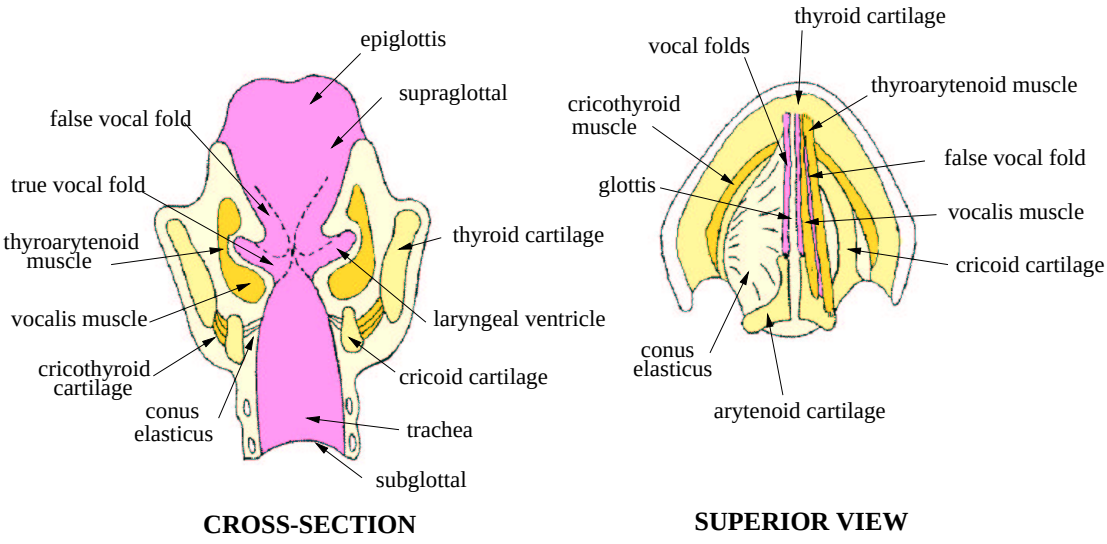


Figure 2.1: *The larynx (after Breen [1])*

Voiced speech, or phonation, is produced by oscillating the fleshy membranes known as the vocal folds. The oscillation is produced by creating a pressure differential between the supra and subglottal regions. When the vocal folds fully occlude the glottis, pressure builds up in the subglottal region creating a pressure differential across the larynx. This forces the vocal folds to open allowing the air to rush out. The flow of air creates a Bernoulli force which, combined with the tension of the vocal

folds, draws them back together. Thus an oscillation is set up with the fundamental frequency being a function of the vocal fold tension which is controlled by the vocalis muscles. This theory of how the oscillations occur is called the *myoelastic/aerodynamic theory of phonation* [1].

The oscillation of the vocal folds modulates the airflow from the lungs producing a sound with a harmonic spectrum which is altered by the positions of the tongue and the velum which redistribute the spectral energy of the sound producing a range of distinct and recognisable sounds. Examples of the vocal tract positions for producing English vowels are shown in Figure 2.2.

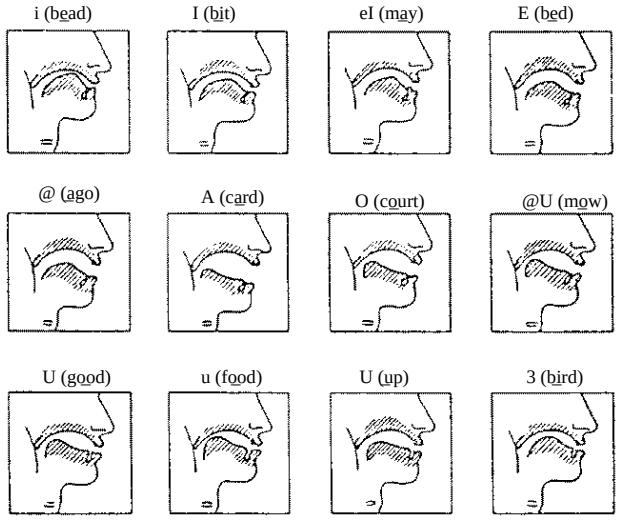


Figure 2.2: *Vocal tract profiles for English vowels (after Fant [2])*

The vocal tract can also affect the sound in another way. By creating a constriction anywhere within the tract, turbulence can be created. This is most commonly seen to occur at the front of the mouth, such as the upper teeth being pushed against the lower lip to produce /v/ as in via. These sounds are called fricatives. If the constriction is a complete closure then pressure builds up behind the constriction, the sudden release of this pressure produces a sudden burst of sound, known as a plosive, such as /b/ in bag. Both the fricatives and the plosives can be created in the absence of phonation. In such cases, for example /T/ in thigh or /p/ in pet, the sound is called 'voiceless'. Examples of vocal tract profiles for both voiced and voiceless fricatives and plosives are shown in Figure 2.3.

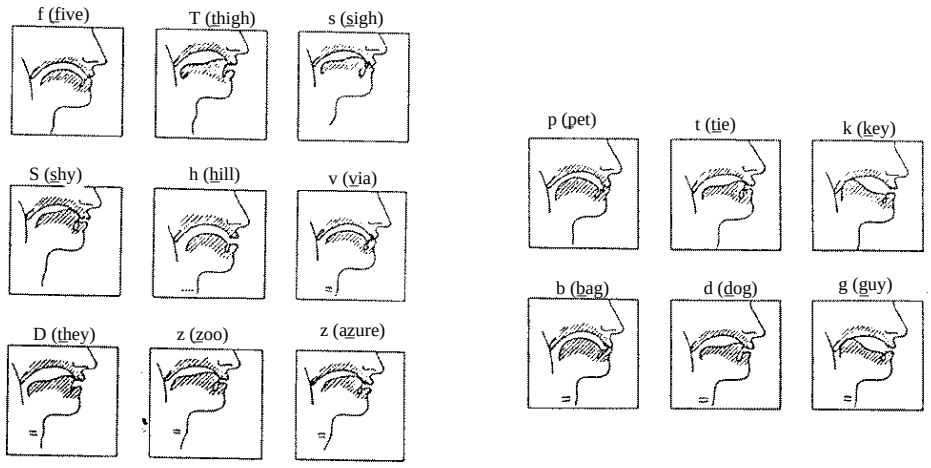


Figure 2.3: *Vocal tract profiles for fricatives and plosives (after Fant [2])*

2.3 Recording

There are two main ways to measure and record speech. Firstly there is the simple acoustic pressure wave recording which is a recording of the pressure at a microphone placed close to the speaker's mouth. Secondly a recording of the glottal closure can be taken using a laryngograph. The laryngograph is a device, placed on the skin outside the larynx, which measures the conductance across the larynx and hence produces a representation of the vocal fold closure cycle. Examples of both these traces are shown in Figure 2.4.

Both of these recordings represent the information in the time domain. Another common representation, of the same recording, which reveals the spectral content of the speech is the spectrogram. The spectrogram is basically a series of power spectra for small, consecutive time sections of the speech signal. The power spectrum for each time interval, generated using a Fourier transform, is represented on the spectrogram as a plot of frequency versus time with the power indicated by the intensity or colour. The width of the time window for the Fourier transform has a great effect on the characteristics of the spectrogram produced. A long time window produces a narrow band-pass filter which allows the harmonic structure to be seen whilst blurring the time definition. A short time window produces a wide band-pass filter and thus blurs the harmonic structure but shows up the time structure corresponding to individual impulses from the glottal closures. Examples of both wide-band and narrow-band spectrograms are shown in Figure 2.5.

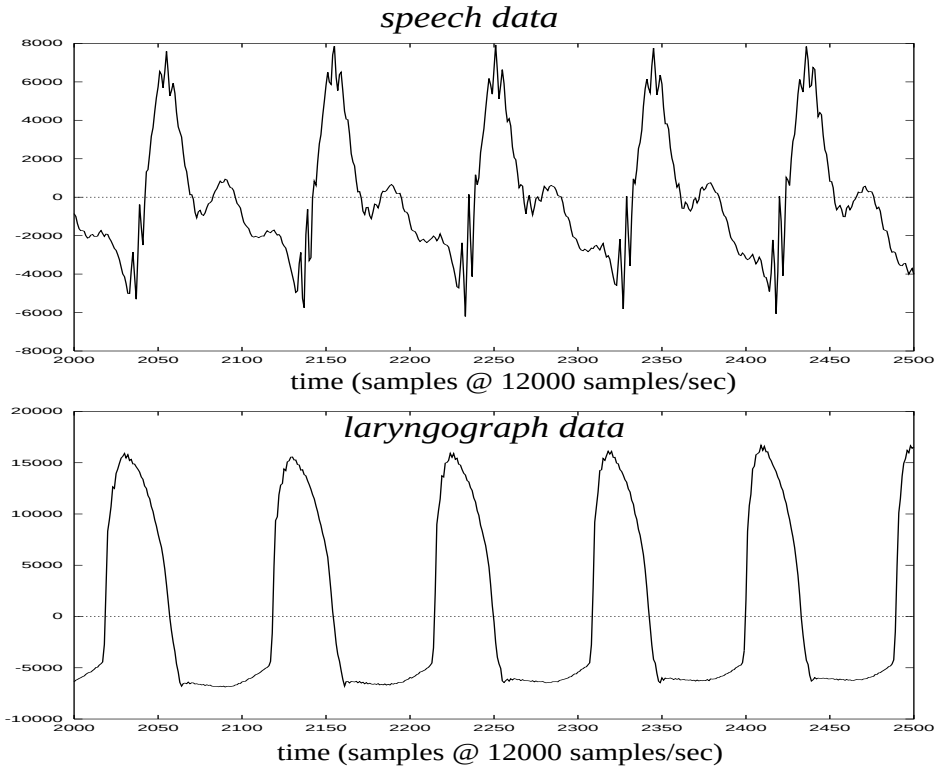


Figure 2.4: *Examples of acoustic wave recording and laryngograph trace*

The spectrogram can be used to identify individual segments of speech, such as phonemes, each of which has its own spectral structure that can become clear to the trained eye. Furthermore the spectrogram can also be used to give information on the frequency of the glottal closure of the signal seen on the wide-band spectrogram as the reciprocal of the time period seen between the vertical lines. The spectrogram can also be used to identify the *formant frequencies* which are the dominant frequencies in the frequency spectrum. Figure 2.6 shows a spectral slice for the vowel /i/ with the formant frequencies marked.

The formants themselves can be used to produce a formant chart which plots F1 against F2−F1 where F1 and F2 are the first two formant frequencies. Such a chart gives a clear indication of how the vowels relate to each other and can therefore be extremely useful when looking for trends in the results.

2.4 Synthesis

Mankind has long sought to be able to reproduce realistic speech from a mechanical or artificial synthesis system. Early examples go back to ancient civilisations attempting

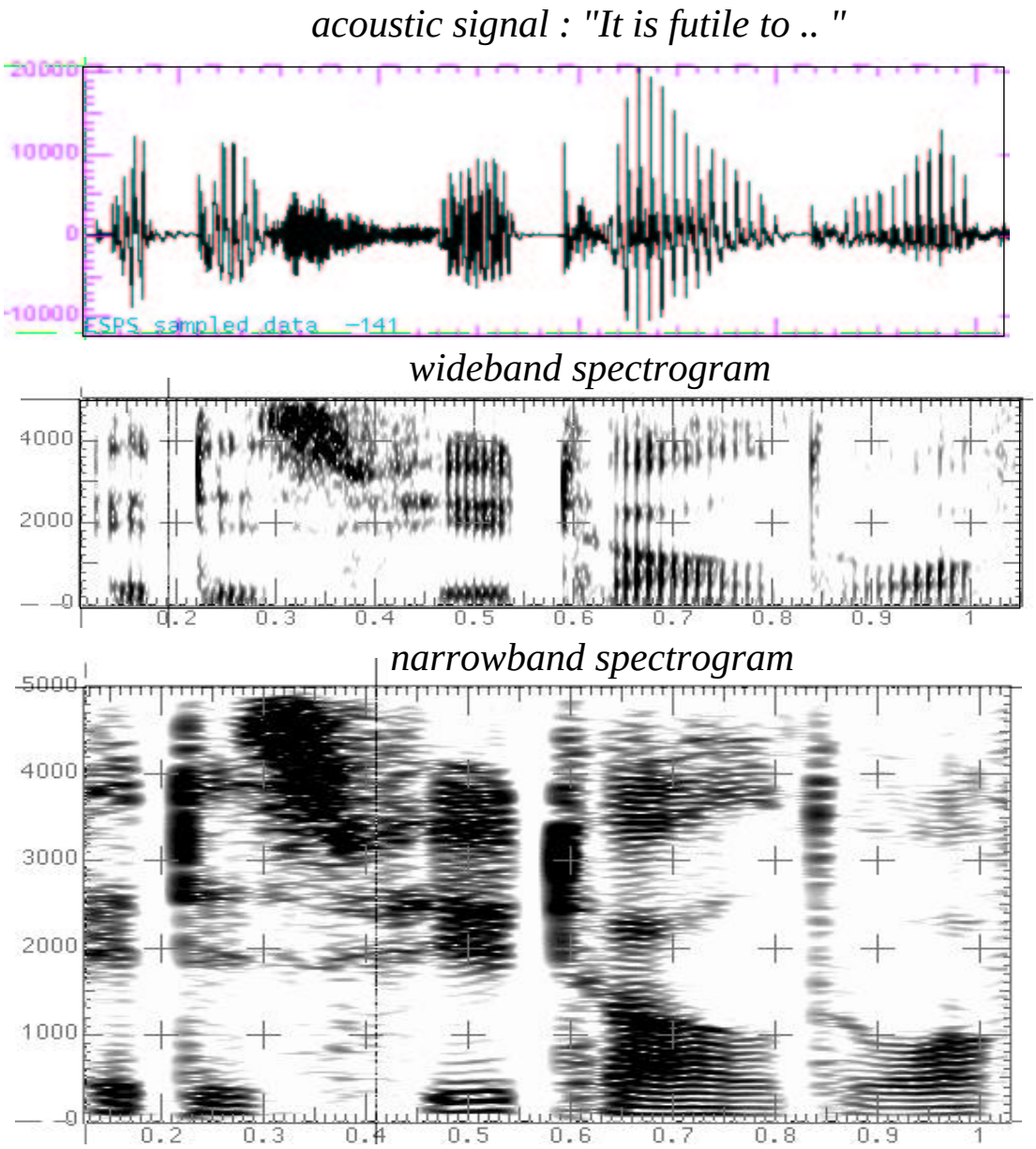


Figure 2.5: *Wide-band and narrow-band spectrograms for speech*

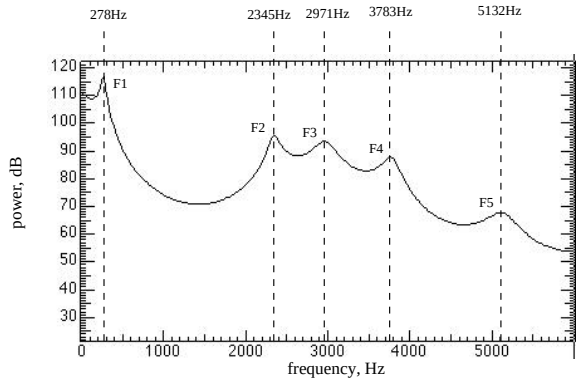


Figure 2.6: *Spectral slice (using LPC with autocorrelation) from vowel /i/ with formants marked*

to make statues of their gods appear to speak through simply speaking into a pipe that is connected to the statue's mouth thus allowing the speaker to be hidden away from the statue itself. Of course this is not real synthesis and it was many centuries before the technology existed so that the first mechanical synthesisers could be developed. One of the earliest examples is the speaking machine produced by Von Kempelen which was an attempt to model directly all the features of the human vocal system using primitive mechanical systems; the lungs are replaced by bellows, the vocal chords by a reed and the vocal tract by a length of rubber tube. Despite its simplicity the machine was reported to be able to reproduce some kind of realistic speech although it required a very skilled operator and therefore again could not be considered to be a complete synthesis system. The advent of electronics brought the electronic analogue of the Von Kempelen machine which was the Voder (VOice DEmonstratoR). The Voder worked by exciting a number of resonators that were controlled by the user in much the same way as playing a piano. It is only really since the advent of micro-electronics and computers that the possibilities of full synthesis techniques, which remove the requirement for a skilled operator, have become a reality.

There are three main ways to synthesise speech :

- attempt to model the physics of the real vocal system as accurately as possible
- reproduce an idealised, if inaccurate, model of the system
- concatenate sections of recorded speech to produce the desired words or phrases.

Each of these broad categorisations covers a plethora of individual techniques and many clever modifications that have been postulated over the years but no single technique seems to have been able to outstrip the rest. What follows is a brief description of some of the most important techniques that exist in these fields.

2.4.1 Accurate real world models

Intuitively the best model of a system has to be the one that relates most closely to the real physics of the actual system. Unfortunately this relies on the fact that the physics of the system are both understood and can be fully quantified. For speech this poses something of a problem; firstly no-one seems to be able to decide exactly what the physics of our vocal system are, and secondly it is extremely difficult to

get measurements of exactly what is happening inside someone's mouth without interfering with their vocal production [33–35]. These problems aside, there are a number of papers that detail each element of the vocal system; the vocal folds and the vocal tract.

The first part of the model is the vocal folds. There are a number of papers detailing the physics of vocal fold oscillation [36] and these lead to a multiple mass model, anything from 2 to 15 masses have been reported [15], but for the model to bare any relation to the real system the complexity has to be enormous making it unwieldy and difficult to use [15]. Modelling of the vocal tract is perhaps even more of a problem; in theory it may be modelled by a tube of variable diameter but this does not take into account the edge effects generated by the flesh around the vocal tract walls which may cause the sound to lose its usual plane wave propagation properties and break down into a number of vortices [16]. Although work has been done to model this effect [2] the models are still very simplistic and do fully reflect reality. It is also necessary for any model of the vocal tract to be able to reproduce the complex three dimensional dynamics of the motion of the articulators. These are but three of the difficulties that are faced by this technique and although there may be scope for this approach in the future there is certainly considerable work still required before it could be used for complete synthesis systems.

2.4.2 Idealised models

Although it is difficult to generate a model that accurately follows the actual physics of the speech production system, it is possible to simplify the problem by attempting to model the output of the system rather than the system itself. This approach is exemplified by the multitude of source/filter synthesis techniques that exist. Basically the idea is that the output can be generated by applying a suitable input signal, or source, to a shaping filter which reproduces the spectral qualities of the desired speech as shown in Figure 2.7. The figure also shows the radiation filter which compensates for the way sound radiates from the lips.

The source/filter approach is very loosely based on the idea that the vocal chords are the source and the vocal tract is a set of acoustic filters but this is clearly somewhat idealised since there is considerable evidence that the operation of the vocal folds is not separable from the action of the vocal tract [37] as is clear from the example of

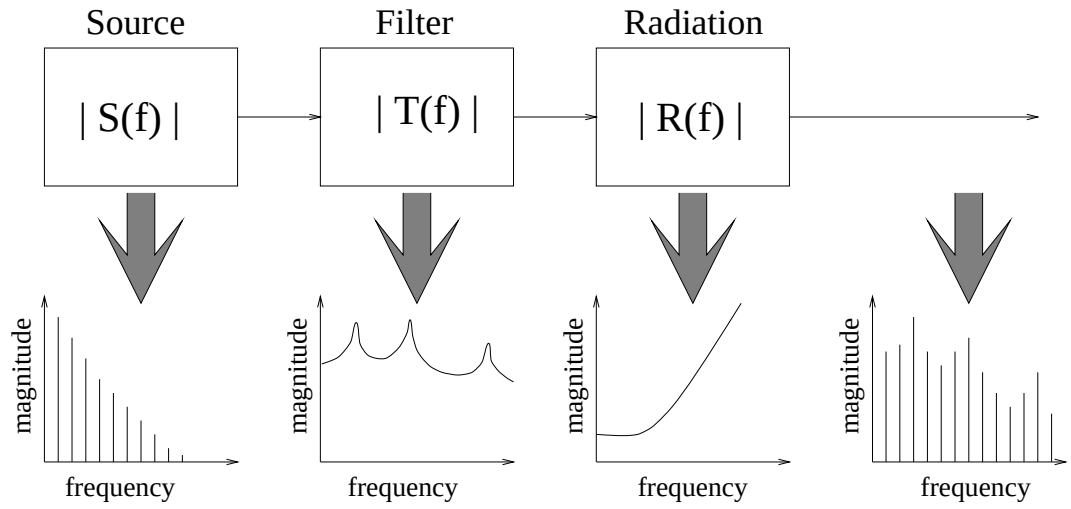


Figure 2.7: *Source/filter approximation for speech.*

open/closed glottal periods: if the vocal folds are open then the length of the trachea must be included in the acoustic cavity model, this additional cavity is not present during the closed period of the glottal cycle, showing a clear connection between the two processes.

The simplest model, for voiced speech, idealises the vocal excitation source to be an impulse train at the specified fundamental frequency. This is a gross over-simplification and results in very poor speech. Better results are achieved by modelling the glottal flow, reconstructed by inverse filtering speech, which is done using approximations such as summed sines and exponentials or expanded/compressed versions of the doubly differentiated volume velocity flow waveform [1]. Many papers have been published theorising as to the importance and structure of the glottal flow waveform [14, 38–40] many of which postulate that for a truly realistic representation then the model must be nonlinear [40]. Some authors have gone as far as to postulate that the shape of the waveform alters the actual style or stress of the voice [41, 42]. Eventually it certainly seems clear that the quality of the synthesised speech will depend greatly on how well the source model itself is constructed [3].

Any source/filter model needs a set of filters to represent the vocal tract. As with the excitation source there are again a number of different approaches to this problem. Simplistically, the overall filter is constructed from a number of bandpass filters that are combined together. This combination can be serial, parallel or in some cases a combination of both. The reason for the variety is that speech is not well defined by a plain serial model which cannot reproduce zeros in the filter which are required in order to produce nasal sounds whereas the parallel structure requires greater control

parameters since each filter requires individual amplitude control. Most modern synthesisers form a compromise and use serial for straight vowel sounds and allow additional parallel branches to provide the necessary flexibility as shown in the Klatt synthesiser given in Figure 2.8.

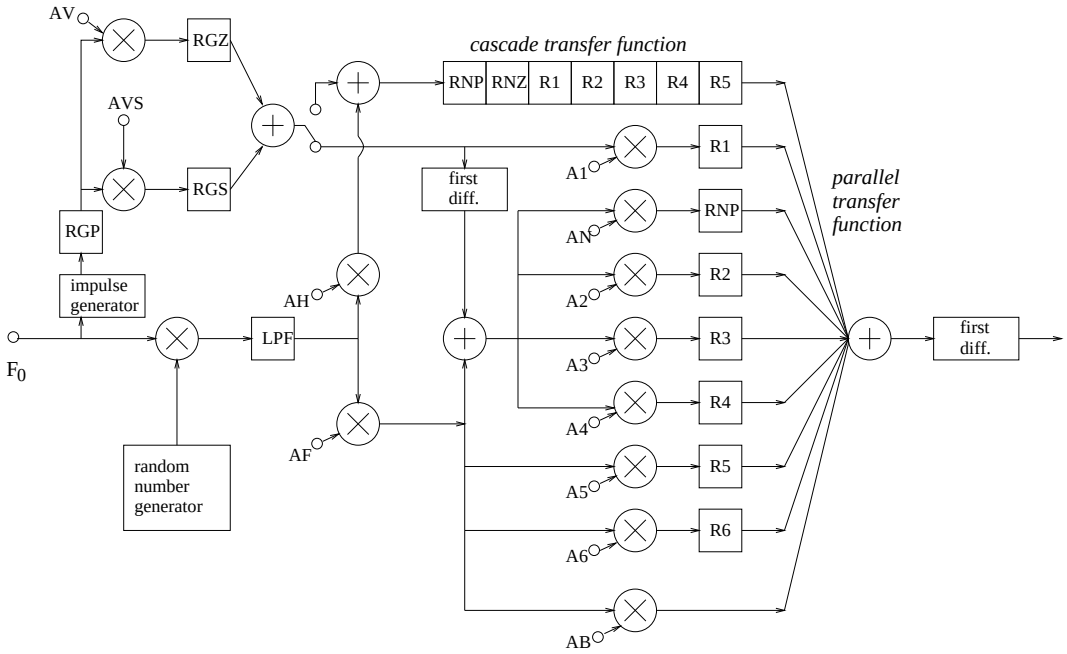


Figure 2.8: *Block diagram of the Klatt synthesiser, from [1], where AH, AF, AVS, AV, AN, AB, A1–6 are amplitude controls and RGP, RGS, RNP, RNZ, R1–6 are resonator and anti-resonator frequency controls.*

So far the discussion has been confined to voiced speech which is not the whole picture. The source/filter approach copes with unvoiced speech by using a noise source to approximate the turbulence caused at the point of frication. This is in fact quite a good model of the unvoiced source although as with the voiced source approximation it is clearly not the full story. Indeed by considering the source to be turbulent airflow then a whole new field of study has been opened up, namely the chaotic analysis of fricatives. Basically the argument is that since there is clear evidence that cavitation noise and simple turbulent models show chaotic behaviour whilst appearing on the surface to be stochastic then perhaps the same holds for the turbulent source in speech. This topic is discussed in more detail later in this chapter.

Overall source/filter models have been extremely successful at producing basic synthesised speech and have been applied to applications such as Text To Speech (TTS) and speech compression through the related field of Linear Predictive Coding (LPC). Unfortunately even though the models do perform well and do produce recognisable

speech they do not produce natural sounding speech. At least not yet anyway. Consequently many commercial synthesisers opt for the last type of synthesis which is the concatenation technique.

2.4.3 Synthesis by concatenation

The obvious alternative to synthesising speech from scratch is to record a database of words that are to be said and then splice, or concatenate, the words together in such an order that they form the desired phrase. This sort of system is extremely limited since the corpus of words required for a complete synthesiser is enormous, most dictionaries have over 80000 words, and most words have a number of different endings, such as singular or plural, and possible intonations such as questioning or asserting. Thus the application of such a synthesiser is limited to small corpus problems such as train time tables or number recital systems, as in the talking clock or phone directories. More general synthesis requires a more general segmentation of speech. This is achieved by using phonemes, as described earlier, to form a database of all the building blocks required to form words in any particular language. It is worth noting that different languages have different phoneme sets and so a synthesiser that works well for one language may be very poor at another. Unfortunately simply splicing two phonemes together does not take into account the coarticulation effects and results in very poor speech. The solution to this is to take a middle ground between recording whole words and recording phonemes : triphones or diphones [6,43] are used. This clearly requires a larger database but does allow the synthesiser to accurately reproduce coarticulation.

Such a synthesiser is again clearly limited by the database of sounds that it contains; the more instances of each sound taken from different contexts the better the speech will be. Full systems are extremely complicated and rely on complex rules to convert the required words into strings of segments from the database. A good example of such a system is the BT Laureate system [5,6,44].

One major problem that any concatenation system has is that it can only really reproduce what it has stored in its database. This means that each time you hear a phoneme you hear exactly the same one. This can become noticeable especially in the extreme example of an extended vowel where the segment is repeated several times. Though there are partial solutions to these problems, such as adding jitter through multiple added sine functions, to date no synthesiser seems to be able to create realistic, natural

human speech.

2.5 Evidence of Nonlinear behaviour

One argument for why synthesised speech sounds unnatural in general is that most synthesisers rely on linear approximations for speech production. There are a number of areas that show evidence of nonlinear behaviour in speech generation:

- Vocal folds
- Turbulent air flow
- Non-plane wave propagation
- Higher order statistics
- Chaotic behaviour

The following sections look at each of these areas.

2.5.1 Vocal folds

There are a number of features about the oscillation of the vocal folds that show they are nonlinear:

- In a linear model the output is proportional to the input and yet the waveform generated from vocal fold oscillation actually changes shape under different amplitude levels. Detailed studies [36, 45] of this effect show that not only does the spectral content of the pulse alter with amplitude but also that the spectral envelope changes with fundamental frequency.
- The vocal chords display bifurcations [46–48]. The clearest example of this is the passage from unvoiced to voiced speech where the oscillations move from an equilibrium state, i.e. not moving, to a pseudo-periodic motion. Bifurcations are a trait of nonlinear systems but in this case there has to be a question as to whether the bifurcations are caused by the driving force passing a threshold, as in classic bifurcation systems, or whether there is some higher muscular force that is controlling the transition.

- Models such as multiple masses routinely include nonlinear coupling between the mass elements [15,48]. This is based on the knowledge that the cartilage and flesh constructing the larynx have nonlinear stretching qualities.
- A number of works have suggested that chaotic modes of operation can be found for vocal fold oscillation. These works should be viewed very carefully since chaos is an extremely difficult phenomenon to quantify, as will become clear throughout this thesis, and can very often cause very misleading results. A full discussion of the work in this field is given in section 2.5.5.

2.5.2 Turbulence

When unvoiced speech is created there is a point of constriction in the vocal tract which causes turbulence to occur. Turbulence is a nonlinear effect which occurs because of an interaction between the air flow and the acoustic sound field. As already discussed there is plenty of evidence to show that cavitation noise [49], which is turbulence, is chaotic and there are a number of works that suggest the same is true of fricative sounds. To keep the flow of the section a full discussion of these works is left for section 2.5.5.

2.5.3 Non-plane wave propagation

The usual model of the vocal tract is that the sound travels along the tract as plane wave propagation. Recently this view has been challenged by Teager and Teager [16] who suggest that the flow consists of a number of vortices. This work is based on examining the air flow at a range of points within the vocal tract using hot wire anemometers. If this is indeed the case then it throws into question the whole acoustic model which is based on the idea that the vocal tract can be considered as a number of acoustic tubes which have well defined reflection and standing wave properties. A good example that shows this effect is given by Kubin [17]: what is the mechanism for human whistling since no part of the vocal tract is in oscillation? The explanation given is, in summary, that an unstable jet of air is created which gives rise to vortices, when the travelling time through the vocal tract matches the frequency of the vortices then periodic vortex shedding occurs at the lips giving rise to the narrow band whistle.

2.5.4 Higher order statistics

Higher order statistics (HOS) can be used to identify the underlying nonlinearities present in a system. Unfortunately the application of HOS theories to noisy signals is very difficult and consequently the application of HOS to speech has not produced conclusive results. However what results have been published [50,51] suggest that there is strong evidence of quadratic phase coupling, which would indicate nonlinearity.

2.5.5 Chaotic behaviour

Several times in this chapter the possible existence of chaotic behaviour has been suggested. This section gives a quick overview of the work that has been conducted and some discussion of the possible shortcomings that may give rise to a number of misleading results.

Most of the work in this field seems to have been inspired by the work of Teager and Teager [16] which gave clear indications that speech was nonlinear; if it is nonlinear then could it be chaotic or fractal in nature?

Maragos [24] suggests that fricatives have a fractal dimension of as low as 1.7 whilst vowels may have a fractal dimension of nearer 1.2. The calculation of this dimension is through the box counting technique [52] which is restricted to a 2 dimensional plane and explains why the figures are so low, and in disparity with the dimension measures given by other authors. It should be noted that this is not saying that the measurements are wrong it is merely pointing out that the box counting dimension looks at the dimension of a waveform not of the generating system itself. The paper also shows calculation of dimension using very small data sets and showing no form of noise cancellation, these are shortcomings that are consistent with many other papers in the field. Both Boshoff [26] and McDowell and Datta [27] give similar analyses suggesting box counting dimensions of between 1 and 2 although McDowell and Datta [27] point out that the accuracy of these results is questionable.

Pickover and Khorasani [53] attempt a similar analysis but on full sentences. This raises the spectre of stationarity; speech is constructed from many small segments that individually may be viewed as stationary, the normal size of these sections is about 10ms which is based on the relatively slow movement of the articulators, but a complete sentence includes many different modes of operation and indeed periods of

complete silence. As a diagnostic tool this approach may have some use if it is used to compare the characteristics of different speakers saying the same sentences, but should not be used to give a definition of the fractal dimension of speech as a whole.

Marcato and Mumolo [54] show that fractal theory can be applied to the LPC to give an efficient coding of the residual signal. Fractals are similarly applied to image coding [55] and speech recognition [56].

McLaughlin and Lowry [25] use the correlation dimension to investigate a range of vowels with the conclusion that although they do seem to show low dimensional properties, the correlation dimension fails to give an accurate measure. These results are consistent with the general disenchantment with correlation dimension when applied to real world signals.

Tishby [18] again examines the correlation dimension giving similar vague reports of dimensions ranging from 3 to 5 for voiced speech. He also looks at the possibility of forming a local nonlinear predictor using neural networks to enhance current predictor based systems. A similar work by Moakes and Beet [20, 57] suggests that speech is low dimensional and they apply Radial Basis Functions (RBF) to both recognition and predictive problems.

Berhard and Kubin [22, 23] give preliminary evidence for low dimensional behaviour, of the order of 1 to 2, for vowels.

In a very recent paper Narayanan and Alwan [28] look at fricatives showing the difficulties of convergence for the correlation dimension but suggesting low dimensions for vowels and high, around 4 to 7, dimensions for fricatives. They also examine the Lyapunov spectra suggesting that vowels have a non-chaotic structure whilst fricatives may have a single positive exponent.

Bohez, Senevirathne and Van Winden [58] give a very clear application of fractal theory to recognition of vowels. Again they do not attempt to infer the actual underlying system's dimension from the fractal dimension but rather use it as a discriminatory tool. In another paper by the same authors they present an analysis of speech using an alternative box counting technique called the amplitude-scale method. Unfortunately this technique seems to give wildly different results from the box counting technique and again is limited to a 2 dimensional space.

Townshend [59] gives a very full overview of the possible uses of nonlinear predictors

in speech along with presenting correlation dimension results of just less than 3. These results again do not appear to be for stationary segments of speech and must be considered with care.

As should be clear there has been considerable work presented in the field although on the whole the problems of noise contamination, data set size and stationarity have not been addressed fully.

2.6 Conclusion

This chapter has given a brief overview of the field of speech with a particular slant towards speech synthesis. The history of the development of speech technology has been given followed by a number of sections detailing the current techniques and synthesis models that are used. Finally a discussion is given of the considerable evidence of nonlinear, and chaotic, behaviour of speech showing that although there has been much research in this area, conclusive results have not been forthcoming although in general they do all point to there being nonlinear, low dimensional behaviour for speech.

Chapter 3

NONLINEAR SYSTEMS AND CHAOS

For many years nonlinear systems have been placed in the back seat whilst the more accessible linear systems led the way. Today the roles are becoming more even with the advent of nonlinear dynamics theory and its natural partner, *chaos theory*. Chaos theory is essentially a science still in its infancy and yet it seems to be bandied around as the all singing explanation for life, the universe and everything. Unfortunately in all the excitement many of the underlying theories have been misinterpreted and misapplied, such as the work by Pickover and Khorasani [53] which uses non-stationary data to name but one example, and the limitations left unsaid. This chapter lays down the background philosophies and building blocks of chaos theory and hopefully casts a gentle warning as to the limitations of its application to the real world.

3.1 History of Chaos

The origins of what has become known as *chaos theory* are difficult to pinpoint in time or to attribute to any one individual person, rather it is better to say that chaos theory has been born out of a multitude of unrelated theories and discoveries from a seemingly unbounded range of disciplines. The earliest stirrings in the field came from the Russian mathematician Lyapunov who investigated nonlinear dynamical behaviour and produced a definition for quantifying the underlying invariant features; namely the sensitivity to initial conditions. His work, even though it is still the cornerstone of any truly complete chaos analysis, became lost amongst the more graphically inclined works that were to follow; Poincaré used topology to show that seemingly complex systems could in fact exhibit regular behaviour; Julia and Fatou found that simple equations could be made to generate infinitely complex geometrical sets, an idea that Benoit Mandelbrot would later clarify and coin the word *fractal* to describe the geometric structure.; Edward Lorenz, in what turned out to be a futile attempt to model weather, produced perhaps the most famous symbol of chaos, the Lorenz attractor. Although these somewhat stunning visualisations of chaos brought the theory to the popular masses it required the works of Feigenbaum, May and Yorke to tie all the ideas

together and provide the underlying principles of a generally applicable theory, indeed it was Li and Yorke [60] who first used the term 'chaos' to relate to this discipline. These principles have since been expounded upon by innumerable researchers, Ruelle [61] and Takens [62] to name but two, to generate the field we now know as chaos theory.

To come right up to date there are a number of very good general tutorial papers available [63–67] which give a solid grounding in the basic underlying theories which have been applied to many modern application areas such as: fractal coding [55]; signal separation [68, 69]; radar signatures [70]; underwater continuous wave signals [71]; chaotic prediction [72–74] and hidden Markov models (HMM) [75].

The following sections in this chapter give a very generalised overview of the elements of chaos theory that are relevant to this thesis and later chapters will deal with the intricacies of real world applications.

3.2 Phase Space and Embedding Dimension

A generalised nonlinear dynamical system can be described by a number of observable output states, for instance the motion of a pendulum could be described using angular position and angular velocity. These outputs, $s_1 \ldots s_m$, describe the overall position in state space of the system as a vector $\underline{s} = \{s_1, s_2, \ldots s_m\}$ which evolves around the state space according to the dynamics described by equation 3.1,

$$\underline{s}(n+1) = f(\underline{s}(n)) \tag{3.1}$$

where $\underline{s}(n) \in \mathbb{R}^m$ is the *state* of the system, n is the time index, and f is a mapping function such that $f : \mathbb{R}^m \rightarrow \mathbb{R}^m$.

An example of such a system is the discrete mapping given in equation 3.2 which describes what is known as the Henon attractor.

$$\begin{aligned} s_1(n+1) &= 1 + s_2(n) - 1.4s_1^2(n) \\ s_2(n+1) &= 0.3s_1(n) \end{aligned} \tag{3.2}$$

Given initial starting states, $s_1(0)$ and $s_2(0)$, the equation can be iterated to produce a series of values for $\underline{s}(n)$ which can be plotted in state space as shown in Figure 3.1. In this case the set of points formed creates an infinitely complex set which never repeats

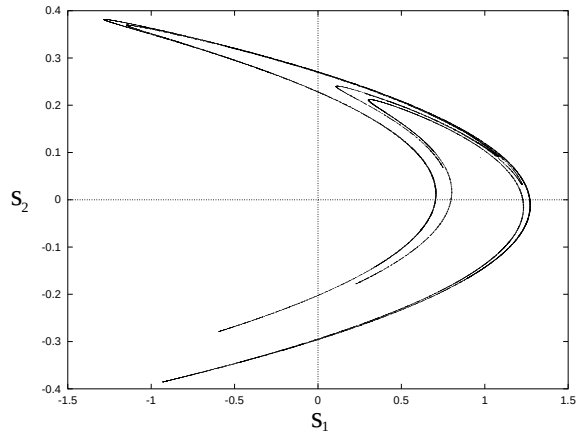


Figure 3.1: *Henon attractor viewed in two dimensional state space*

or overlaps. In general this is not the case and it is better to step back to a much simpler system in order to describe what is meant by the attractor of the system.

A simple pendulum, using angular position and angular velocity as the observable states, has a *point attractor*; all initial states will converge onto a single point in state space as shown in Figure 3.2(b). By modifying the system so that the pendulum has a driving force, the system can be forced into a simple oscillation which has a *periodic attractor* as shown in Figure 3.2(c). Adding a second base frequency of oscillation

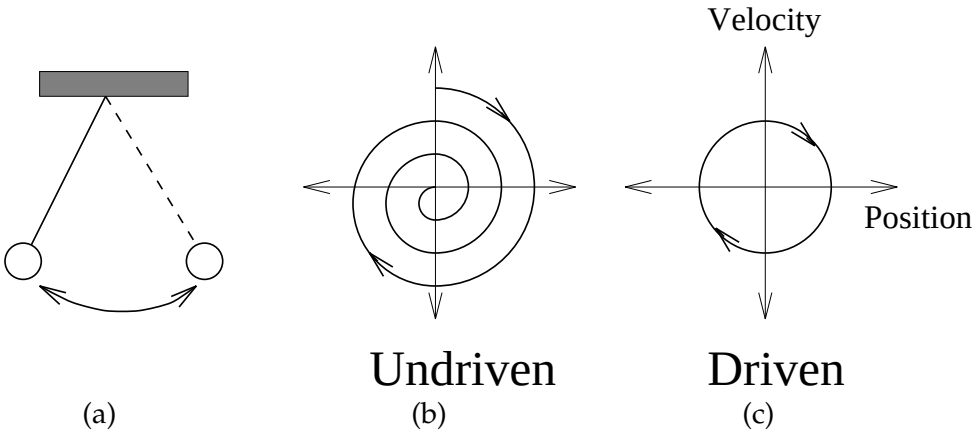


Figure 3.2: *(a) the pendulum (b) undriven (c) driven*

generates an attractor that requires three dimensions to allow visualisation. In this case the attractor is a two torus, or *2-periodic*, as shown in Figure 3.3. In Figure 3.3 the base frequencies ω_1 and ω_2 are commensurate, that is ω_1/ω_2 is rational, and consequently the attractor repeats itself with a periodic structure. If the frequencies are incommensurate then the attractor, that is the set of possible states of the system, becomes a manifold defined as the entire surface of the torus.

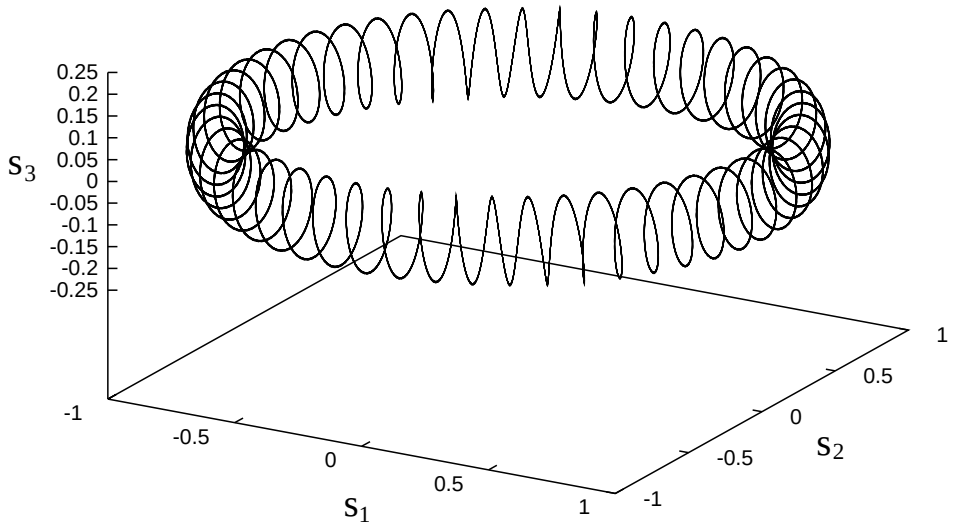


Figure 3.3: *The attractor of a 2-periodic system*

Chaotic systems also define a set of trajectories that cover the surface of a manifold in state space. Unlike the simple pendulum though, they exhibit a very important characteristic which sets them apart from other non-chaotic attractors; the trajectories exhibit a combination of both convergence **and** divergence in the state space. This can be seen by a comparison of the 2-torus with a well known chaotic system such as the Lorenz system as given in equation 3.3 using $\sigma = 16.0$, $r = 40.0$ and $b = 4.0$.

$$\dot{X} = \sigma(Y - X) \qquad \dot{Y} = rX - Y - XZ \qquad \dot{Z} = -bZ + XY \tag{3.3}$$

In the periodic case, if you trace two nearby points around the attractor then you find that the separation of the points remains approximately constant. The Lorenz system however shows far more complex behaviour. If two points are traced around the Lorenz attractor then even after only a short time they can become totally separated, that is they have shown divergence. This can be seen in Figure 3.4. At the same time it is also clear that there must be a contraction, or convergence, occurring since the trajectories never leave the overall attractor manifold. This combination of contraction and divergence is very important in chaotic systems and a full discussion of this quality will be returned to later in the thesis. For this chapter the important consequence of this behaviour is that, even though the attractor lies on a manifold in the state space, it does

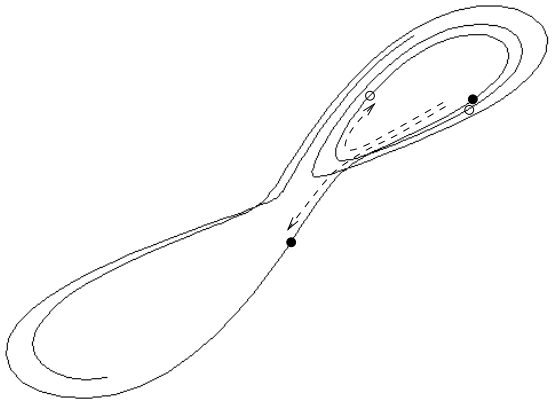


Figure 3.4: *Divergence of points on the Lorenz attractor*

not cover every point on that manifold, that is to say that two different trajectories may pass infinitesimally close to one another but may never cross or touch. In order for this property to be observed, the system must be embedded into the correct embedding dimension. This idea should be clear from the simple example of the 2-torus again. In two dimensions, the dimension of any picture on a piece of paper, the trajectories can be seen to cross at many points around the attractor as shown in Figure 3.5. The

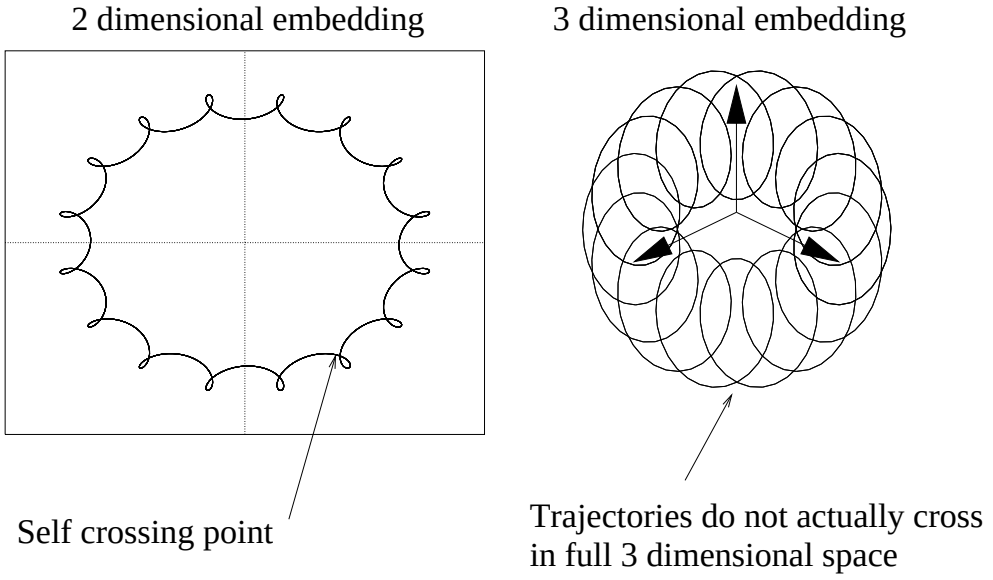


Figure 3.5: *Self crossing on a two dimensional torus*

figure also shows the 3-dimensional embedding of the same torus. Clearly because of the printing medium the trajectories still appear to cross but if the figure could be viewed in its full three dimensions then these crossings would no longer appear.

Such points that appear to be crossings in one embedding but disappear in a higher embedding are called *false near neighbours* [76]. Takens [62], and later Sauer, Yorke and Casdagli [77], defined a theory that allows us to place a theoretical sufficient limit on the embedding dimension required to *ensure* that no self crossings occur. In essence the argument is that in a space of dimension d_e, a manifold of dimension d1 and a manifold of dimension d2 will intersect in a manifold of dimension $D_i = d1 + d2 - d_e$. For instance, in 3D space a surface of dimension 2 and a line of dimension 1 intersect at a point (dimension zero), similarly two surfaces intersect at a line.

So for *self intersection* of a manifold of dimension d, $D_i = 2d - d_e$ and therefore in order to ensure no self crossings, that is to say $D_i = -1$, we have the sufficient condition

$$d_e >= 2d + 1$$

In practice this is not a hard and fast rule and there are many applications where lower embeddings are suitable or indeed are required. For the purposes of this chapter it is sufficient that the reader understands that an embedding lower than the theoretical d_e will in all likelihood produce self crossings and these must be considered in any analysis that is based on that embedding.

So far we have discussed embedding in terms of a number of observable output states of a system. It is also possible, and indeed often a necessity, to construct the embedding from only one observable state or time series. The most common technique is that suggested by Takens which uses the method of delays to embed a single time series into a d_e dimensional space.

Simply stated, Takens' method involves moving a window of length m through the data, and taking each snap-shot seen in the window as a row of a m-column matrix. That is, for a time series

$$x(t) = (x_0, x_1, x_2, x_3, \ldots x_i, \ldots) \tag{3.4}$$

the reconstructed trajectory matrix takes the form

$$\mathbf{X} = \begin{pmatrix} x_0 & x_1 & x_2 & \dots & x_{m-1} \\ x_1 & x_2 & x_3 & \dots & x_m \\ x_2 & x_3 & x_4 & \dots & x_{m+1} \\ \vdots & & & & \end{pmatrix}. \tag{3.5}$$

Takens showed that such a matrix fully describes the geometrical properties of the general dynamics of the original m-dimensional system with no self intersection of trajectories. Careful choice of the delay time, i.e. the sampling period between successive values of x_i, improves the results by, in effect, opening up the attractor. A formal technique known as mutual information is often used to provide an estimate of the optimal delay between samples. Mutual information, $I_m(S, Q)$, measures how much the uncertainty, $H(S)$, of a measurement q is reduced by making the measurement s. Hence $I_m(S, Q)$ can be defined

$$I_m(S, Q) = H(Q) - H(Q|S) = I_m(Q, S) \tag{3.6}$$

where H is defined as

$$H(S) = -\sum_i P_s(s_i) \log P_s(s_i) \tag{3.7}$$

and

$$H(Q|S) = -\sum_{i,j} P_{sq}(s_i, q_j) \log[P_{sq}(s_i, q_i)/P_s(s_i)] \tag{3.8}$$

where S denotes the system, s_i denotes a measurement of S and $P_s(s)$ is the probability density at s.

By substituting $[s, q] = [x(t), x(t+\tau)]$ we get the mutual information for two points separated by τ. Fraser and Swinney [78] suggest that the optimum τ to use is the value for which $I_m(x(t), x(t+\tau))$ reaches its first minimum. Figure 3.6 shows mutual information for Lorenz data. The first minimum occurs at $\tau \approx 10T_s$ where $T_s = 0.01$.

More complex techniques of embedding do exist and where data is contaminated by noise then the method of delays is often no longer sufficient. The actual technique used to produce the embedding is not important so long as a full embedding has been produced and therefore for the purposes of this chapter it is not worthwhile complicating matters. A full discussion of embedding is undertaken in the next chapter.

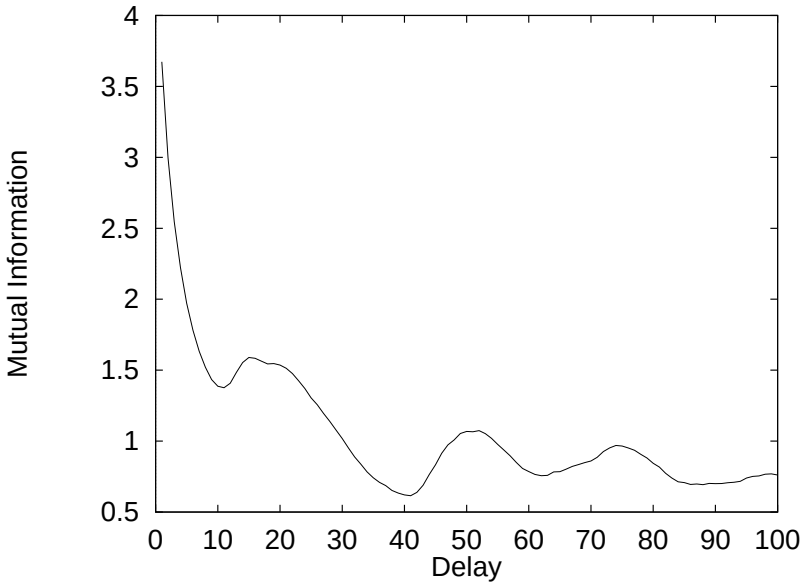


Figure 3.6: *Mutual information for Lorenz data*

3.3 Non-integer Dimensions

As has already been alluded to, a chaotic attractor does not entirely fill its state space. If we take a commensurate torus and zoom into the picture then eventually the trajectory appears to be a one dimensional line in state space as shown in Figure 3.7. Similarly

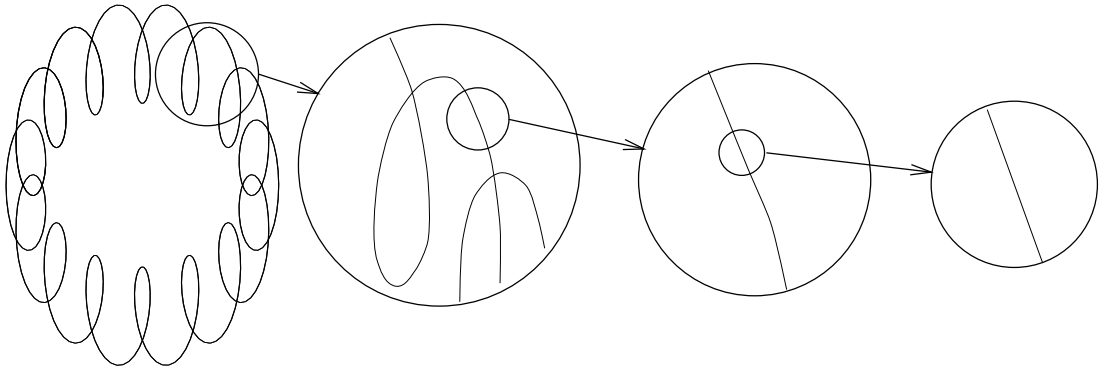


Figure 3.7: *Zooming into a torus until it becomes a line*

an incommensurate torus would have resulted in a flat two dimensional plane. A chaotic attractor is rather different. Zooming into the Henon map, Figure 3.8, results in revealing more and more self-similar complexity[1]. Indeed no matter how much the map is magnified the complexity of the structure is apparent. It is this rather strange and fantastic quality that has led to such attractors being called *strange attractors*.

[1] it should be noted that this is a comparison of a flow with a map but the idea of increasing complexity can still clearly be seen

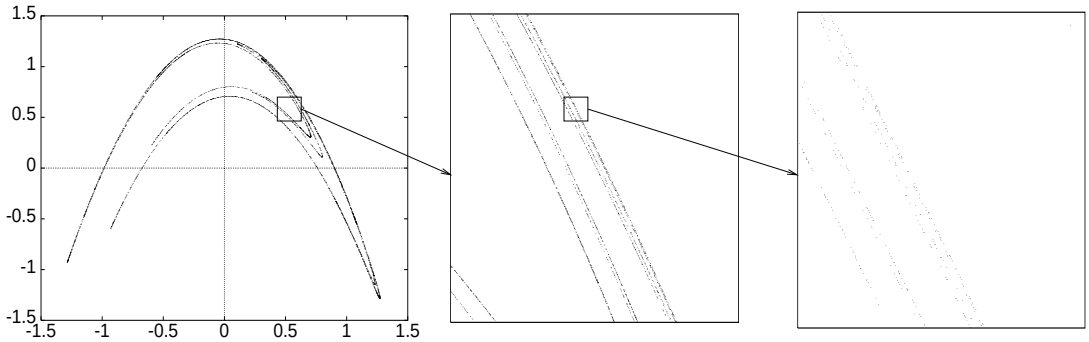


Figure 3.8: *Zooming into the Henon map reveals new levels of complexity*

Such a structure does pose something of a problem though. The attractor does not fill the whole of the 2 dimensional space and yet does not break down into a 1 dimensional structure, hence should the structure be called 1 or 2 dimensional? The obvious answer is "somewhere in between". This is the solution that Mandelbrot [79] called the fractional dimension and what has now become more commonly known as the fractal dimension of the attractor. There are a number of different ways to calculate such a dimension and each technique actually calculates a theoretically different quality of the attractor. Because of this all the techniques give different values and therefore it is important to know which measures exist and what they actually signify.

There are several different techniques that can be used to measure the fractal dimension [80, 81]. Four common approaches are:

- capacity dimension,
- information dimension,
- correlation dimension,
- lyapunov dimension.

The capacity dimension is based on how many d dimensional hypercubes would be needed to cover the attractor. If a line of length L can be covered by $N(\varepsilon)$ line segments of length ε then

$$N(\varepsilon) = L/\varepsilon.$$

A two dimensional square of side L requires $N(\varepsilon) = L^2(1/\varepsilon)^2$ squares to be covered.

For increasing dimension, d, the argument can be generalised to

$$N(\varepsilon) = L^d(1/\varepsilon)^d.$$

If ε is taken to be small then, taking logarithms and rearranging, the definition for the capacity dimension, d_{cap}, can be produced as in equation (3.9).

$$d_{cap} = \lim_{\varepsilon \to 0} \frac{\log N(\varepsilon)}{\log(1/\varepsilon)} \tag{3.9}$$

The information dimension gives a measure of how fast the information required to specify any point on the attractor increases as the size of ε decreases. Thus the information dimension, d_i, is defined as

$$d_i = \lim_{\varepsilon \to 0} \frac{I(\varepsilon)}{\log(1/\varepsilon)} \tag{3.10}$$

where $I(\varepsilon)$ is defined by information theory to be

$$I(\varepsilon) = -\sum_{i=1}^{N(\varepsilon)} P_i \log \frac{1}{P_i} \tag{3.11}$$

and P_i is the probability that a point is within the ith box if the attractor is covered by $N(\varepsilon)$ boxes of size ε.

The correlation dimension is a measure of how the number of neighbours, points within distance ε of each other, vary with decreasing ε. The usual definition for d_c is

$$d_c = \lim_{\varepsilon \to 0} \frac{\log C(\varepsilon)}{\log \varepsilon} \tag{3.12}$$

where $C(\varepsilon)$ is the correlation function given by

$$C(\varepsilon) = \lim_{N \to \infty} \left[\frac{1}{N^2} \sum_{i,j=1}^{N} \delta(\varepsilon - \| \underline{s}_i - \underline{s}_j \|) \right] \tag{3.13}$$

and $\underline{s}_i$ and $\underline{s}_j$ are points on the attractor, $\delta()$ is the Heaviside step function and $\| \|$ is the Euclidian norm.

In order to describe the Lyapunov dimension it is first necessary to explain what is meant by Lyapunov exponents. Since these are explained at length later in the chapter we will leave that discussion of Lyapunov exponents until then, for now suffice to say

that the Lyapunov dimension is defined as

$$d_L = j + \frac{\lambda_1 + \lambda_2 + \ldots + \lambda_j}{|\lambda_{j+1}|} \tag{3.14}$$

where λ_i is the ith Lyapunov exponent.

All of these measures provide a theoretically geometrical invariant for an attractor although some are more useful than others when applied to the real world. In particular the probabilistic nature of the correlation and information dimensions make them practical to apply to limited length data sets.

3.4 Predictability

The ability to predict the future states of a system given an arbitrary set of starting positions is intrinsically linked to the idea of chaos. As has already been stated a chaotic system will exhibit a combination of divergence and convergence within the phase space. This combination leads to the possibility that two similar starting positions will evolve to radically different evolved positions which are still on the attractor, as shown in the example in Figure 3.4. This property is known as *sensitivity to initial conditions* and places fundamental limits on the predictability of a system. A good example of this is the classic weather scenario used in the infamous butterfly analogy : a butterfly flapping its wings over India may have the effect of creating a whirlwind over Texas. Clearly the analogy is somewhat fanciful but what it is trying to convey is the idea that a small effect that alters the start position, the disturbance created by the wings, will cause the system to diverge along a completely different path perhaps even following a pattern that has a whirlwind in Texas. Clearly the system is predictable over the short term, a weather forecast for the next few hours is rarely too far removed from reality, and yet forecasts for several days ahead can often be totally wrong showing that long term predictability is low. An important property to note though is that the forecast could say "the weekend will have weather" and it would be right; this is showing that although the system is not predictable in an exact sense, it can be predicted that the system will not leave its attractor. This combination of short term predictability and long term unpredictability can be used as an initial pointer for identifying a chaotic system. Just such a technique is employed by Sugihara and May [82] on measle epidemic data and Casdagli [83] on a range of different real world data sets.

3.5 Lyapunov exponents

The Lyapunov exponents of a system are a set of invariant geometric measures which describe, in an intuitive way, the dynamical content of the system. In particular, they serve as a measure of how easy it is to perform prediction on the system. When talking about a *system* here, it is easiest to think of a single configuration of the system which has a particular set of trajectories in phase space, i.e. an attractor.

Lyapunov exponents quantify the average rate of convergence or divergence of nearby trajectories, in a global sense. A positive exponent implies divergence, a negative one convergence, and a zero exponent indicates the temporally continuous nature of a flow. Consequently a system with positive exponents has positive entropy, in that trajectories that are initially close together move apart over time. The more positive the exponent, the faster they move apart. Similarly, for negative exponents, the trajectories move together. A system with both a positive and a negative Lyapunov exponent is said to be chaotic.

Mathematically, the Lyapunov spectrum ($\lambda_{i=1..k}$) can be defined by:

$$\lambda_i = (\lim_{n\to\infty} \frac{1}{n} \ln(\mathrm{eig} \prod_{p=0}^{n} J(p)) \tag{3.15}$$

where J is the Jacobian of the system as p moves around the attractor. As such, it can be seen that the Lyapunov exponents describe the average rate of exponential change in the distance between trajectories, in a set of orthonormal directions within the embedding space.

If the differential equations defining the system are known, then there are established techniques [84] for applying this formula, and the entire Lyapunov spectrum can be calculated.

Unfortunately, the differential equations describing a process, particularly a measured "real world" process, are not generally known, and a method for extracting Lyapunov exponents directly from a time series must be resorted to. A number of algorithms have been proposed to address this [29, 49, 49, 84–92], but problems have been found with successfully applying all such algorithms known to the author. Specifically their performance in the presence of noise and short data sets degrades significantly as would be expected since short data sets do not provide a dense enough coverage of

the attractor to maintain linearity approximations. One of the problems often faced in real world situations is that the data is only obtainable in short bursts, such as small chaotic regions during transition states, which are usually contaminated with some level of background noise. In the following chapter a novel approach is presented to overcome these difficulties by utilising both a series of short data sets to produce a composite attractor for the system and a noise reduction technique.

3.6 Application to the Real World

It is important to stress that the bulk of this chapter has been a brief overview of the techniques and ideas that exist in the chaos community. As with most overviews of this field [63, 93–95] it is a somewhat idealised view that is presented. When applied to real-world situations many of the techniques simply do not work as expected; in particular constraints on data set size [96–102] and the inevitable background noise problems can give rise to misleading and often contradictory results. Many authors have alluded to these problems and in some part proposed solutions, although there seems to be a fairly universal distrust of the 'canned' routines that can be found. Thus it could be postulated that the most important parts of any analysis using the ideas of chaos must be firstly, getting to know the exact behaviour of the algorithms used, and secondly getting a real overall feeling for the data through using a number of different analysis techniques.

3.7 Summary

This chapter has laid out the basic building blocks and philosophy behind chaos theory and should serve as an introduction to the detailed explanations and derivations that follow in the next chapter. Particular reference has been made to the problems that occur when translating what is still a science in its infancy into the real world arena where the amount of available data is limited and where noise plays a centre stage role.

Chapter 4

NONLINEAR ANALYSIS TOOLS

With many subjects there is a huge chasm between the theoretical and the practical application. Chaos theory is particularly prone to this, especially since the two prime antagonists to the chaotician are noise and shortage of data, both of which are inherent properties of real world data. This chapter details the chaos analysis tools that will be used later in the thesis presenting both algorithms and details of application. Particular reference is made to overcoming the 'real world' problems of noise and data set size, especially for the Lyapunov spectra extraction algorithm which details a new and innovative noise reduction technique. The tools given are

- Time delay and SVD embedding
- Correlation dimension
- Singular value analysis
- Local singular value analysis
- Lyapunov spectrum
- Short term prediction

4.1 Time Series Embedding

A time series measurement from a system is, loosely speaking, a scalar measurement of the vector process which is the system. The initial problem, therefore, is to reconstruct the full dynamics of the system from that scalar measurement. The most common way of doing this is to use Takens' method of delays [62] to embed the time series into a m dimensional space which contains a smooth manifold for the d dimensional system where $m \geq 2d + 1$.

Simply stated, Takens' method involves moving a window of length m through the data, and taking each snap-shot seen in the window as a row of a m-column matrix.

That is, for a time series

$$x(t) = (x_0, x_1, x_2, x_3, \ldots x_i, \ldots) \tag{4.1}$$

the reconstructed trajectory matrix takes the form

$$\boldsymbol{X} = \begin{pmatrix} x_0 & x_1 & x_2 & \ldots & x_{m-1} \\ x_1 & x_2 & x_3 & \ldots & x_m \\ x_2 & x_3 & x_4 & \ldots & x_{m+1} \\ \vdots & & & & \end{pmatrix}. \tag{4.2}$$

Takens shows that such a matrix exhibits the general dynamics of the original m-dimensional system. Careful choice of the delay time, i.e. the sampling period between successive values of x_i, improves the results by, in effect, opening up the attractor.

For 'clean' data, this method is often sufficient, however it takes no account of noise on the signal and therefore when used in a noisy environment it produces a noisy attractor. The method of singular value decomposition (SVD) reduction, described by Broomhead and King [85, 103], addresses this problem. The data is projected onto a phase space defined by the singular vectors of the data, which can then be partitioned into a signal subspace and a noise subspace. This technique is discussed in detail by Gibson et al [104] who show that SVD is the optimal embedding in the signal to noise ratio sense subject to constraints on the noise being stationary, symmetric and additive.

A time delay embedding is first carried out, producing a $N \times w$ trajectory matrix $\boldsymbol{X}$, where the window length w (i.e. the number of columns in $\boldsymbol{X}$) is chosen to be much greater than the expected dimension of the system. This matrix is then factorised into its singular values[1]

$$\boldsymbol{X} = \boldsymbol{S}\Sigma C^T \tag{4.3}$$

where Σ is a diagonal matrix containing the singular values of $\boldsymbol{X}$ with the ordering $|\sigma_1| \geq |\sigma_2| \geq |\sigma_3| \geq \ldots \geq |\sigma_w| \geq 0$, and $\boldsymbol{S}$ and $\boldsymbol{C}$ are matrices of the singular vectors associated with Σ. The singular vectors $\underline{s}_i$ comprising $\boldsymbol{S}$ are the eigenvectors of the *structure matrix*, $\Theta = \boldsymbol{X}\boldsymbol{X}^T$. The singular vectors $\underline{c}_i$ comprising $\boldsymbol{C}$ are the eigenvectors

[1] The singular values can be calculated using any of the available algorithms although it has been found the Numerical Recipes [105] routine **svdcmp.c** to be sufficient. It should be noted that the technique used needs to be able to cope with large window lengths and this should be considered when choosing a routine.

of the *covariance matrix*, $\Xi = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$. The singular values in Σ are the square roots of the eigenvalues of either Ξ or Θ (the structure matrix and the covariance matrix have the same eigenvalues). Rewriting Equation 4.3,

$$\Sigma = \boldsymbol{S}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{C} \tag{4.4}$$

facilitates the calculation of these singular values.

Since the singular values contained in Σ are the root mean square projections onto the basis vectors, the number of non-zero values in Σ should give the number of degrees of freedom in $\boldsymbol{X}$. Noise on the data, however, will generate spurious degrees of freedom, evident as extra non-zero values in Σ.

It should be noted that this process is dependent on choosing an appropriate sampling rate for the data (generally taken as the Nyquist frequency). Since chaotic systems don't have a clear cut-off frequency then the choice of sampling rate is not well defined and is yet another grey area in chaotic analysis. Also, for data generated artificially, from a set of differential equations for example, no noise floor will be apparent since the precision of the data in the time series will be dependent on the precision of the calculation.

The embedding space has now been partitioned into a d-dimensional signal subspace and its orthogonal complement, the $(w - \mathrm{d})$-dimensional noise subspace. The dynamical information of interest is contained within the signal subspace, so it is now desirable to reduce the w-dimensional trajectory matrix $\boldsymbol{X}$ to a d-dimensional reduced trajectory matrix $\hat{\boldsymbol{X}}$. As with any technique in this field, if the noise floor is higher than amplitude variations due to the system dynamics in one time step, then this dynamical information will be thrown out with the noise.

The reduced trajectory matrix $\hat{\boldsymbol{X}}$ is calculated from

$$\hat{\boldsymbol{X}} = \boldsymbol{X}\,\mathrm{C}_{(\mathrm{d})} \tag{4.5}$$

where $\boldsymbol{C}_{(\mathrm{d})}$ consists of the first d columns of $\boldsymbol{C}$. Remember that $\boldsymbol{C}$ contains the ordered eigenvectors of the covariance matrix Ξ. $\hat{\boldsymbol{X}}$ is thus an $\mathrm{N} \times \mathrm{d}$ matrix, containing a reduced trajectory representing the noise-reduced dynamics of the system under study, where N is the number of points on the trajectory, and d is the perceived dimension of the phase space in which the attractor lies.

4.2 Dimension

When talking about the dimension of a system or an attractor it is easy to become confused. The confusion arises because it has become common to mix up the dimension of a system with the embedding dimension of a system. Put simply the dimension defines the number of degrees of freedom that the system has which is nearly always different from the required embedding dimension, as already discussed in the previous chapter. Since a sufficient value can be placed on the embedding dimension, through Taken's theory, from knowledge of the system dimension but not vice versa then it is sensible to look for the system dimension rather than the embedding dimension. It is quite a common mistake in fact to attempt to infer something of the system dimension from the embedding dimension and so techniques such as false nearest neighbour analysis [76] need be carefully applied and kept in context. This section lays out a number of techniques that can be used to determine the system dimension.

4.2.1 Correlation Dimension

Calculation of the correlation dimension is a well documented subject with a number of associated 'canned' routines being available. The most common of these routines is the GRASS suite of progroms which is based on the Grassberger and Procacia algorithm [95]. In detailed tests GRASS operates extremely well on completely noise free data, however as shown in Figure 4.1 the convergence of the results is seriously affected by the introduction of even small levels of additive Gaussian noise.

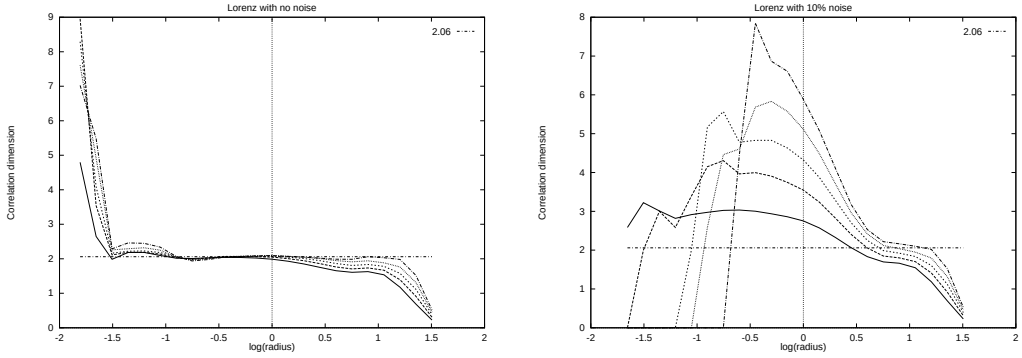


Figure 4.1: *Correlation dimension for Lorenz data with no additive noise and for additive Gaussian noise at 10% of the signal variance.*

This noise degradation is noted by several authors [96–98, 100, 101, 106] who also

comment on the importance of data set size to ensure that the attractor is sufficiently defined. Unfortunately there does not seem to be a realisable solution to the effects of noise and therefore since other techniques do exist that can yield similar information then it seemed more sensible to push with the alternative approaches, as given in the following sections.

4.2.2 Singular Value Decomposition Spectra

Singular value decomposition (or principal component analysis (PCA)), as already described in section 4.1, defines the power content in each of a set of orthogonal axes which are defined as linearly independent directions. In theory an attractor should exhibit a singular spectrum which has a number of significant singular values, one for each degree of freedom for the system, and a number of singular values which lie on a noise floor [85]. This basic idea can be seen using the example of the Lorenz system as shown in Figure 4.2. Lorenz has three degrees of freedom and three singular values can be seen above the leveling off point which is usually taken to be the noise floor. It should be noted that the placement of the noise floor is quite arbitrary and needs to considered with care. Unfortunately SVD analysis does have a number of problems.

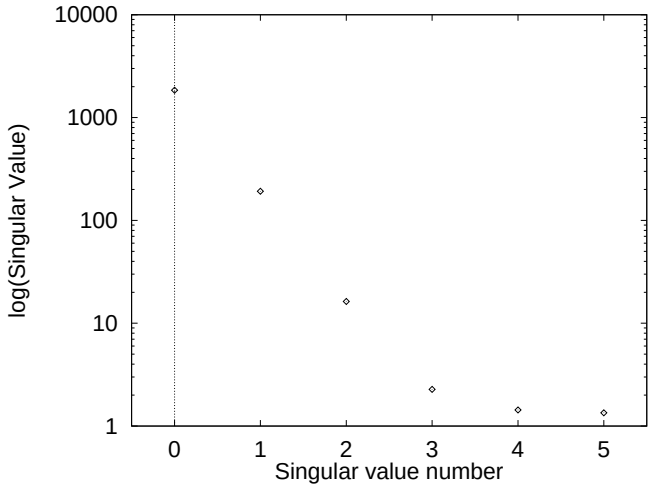


Figure 4.2: *SVD for Lorenz*

Firstly it is a global technique: it is common for attractors to exhibit variable levels of dimension according to the exact state space position [83] and therefore it is instructive to look at small sections of the attractor and build up an overall feel for the dimension of the system. Secondly it is linear: chaos occurs in nonlinear systems and as such a globally linear technique can be prone to providing misleading results when applied to such systems [107].

Fortunately all is not lost since it is possible to extend the basic ethos of the technique, investigation of linearly independent axes, by looking at local SVD analysis as described in the next section.

4.2.3 Local SVD Techniques

The analysis takes a singular value decomposition of a neighbourhood set centred on one test point in the attractor. As the neighbourhood size is increased, so the eigenvectors of any non-noise basis vectors should scale accordingly whilst the noise dominated basis vectors remain flat. The number of rising values which have a gradient of one correspond to the number of degrees of freedom of the system. This is complicated by the fact that where the system does not have an integer dimension then one degree of freedom will not be seen: the attractor does not completely fill the space for that dimension and therefore does not scale with gradient one. This is similar to the discussion of information dimension where the dimension of the system is given by the next highest integer. Thus this analysis can be used to say that the number of degrees of freedom, or dimension, of the system is equal to either n or $n+1$ where n is the number of values with a gradient of one.

Another problem with this technique is that other scaling factors become apparent if there is curvature of the manifold on which the attractor sits. To see this it is easiest to examine what happens for a single frequency orbit, a commensurate 2-torus and an incommensurate 2-torus. Figure 4.3 shows the three attractors along with the plots of the singular spectra. For the single frequency orbit there is one value rising with gradient one, signifying that the trajectory is 1 dimensional (a line), and another value that has gradient of 2 which signifies that there is a curvature of the manifold; you can imagine taking a long line and bending it into a circle that repeats itself, the circle must therefore still be of the same dimension as the line itself.

A 2-torus can be formed by adding two sinusoids and therefore the system should have 2 degrees of freedom. A commensurate 2-torus is a special case where the constraint on the frequencies of the sinusoids means that only one degree of freedom actually exists. This can be seen from the local singular value analysis. Looking at the commensurate 2-torus one could be forgiven for thinking that the attractor is three dimensional. In fact since the trajectory repeats itself on each cycle, the torus does not fill the three dimensional space but rather is still just a curved version of the original line. This is

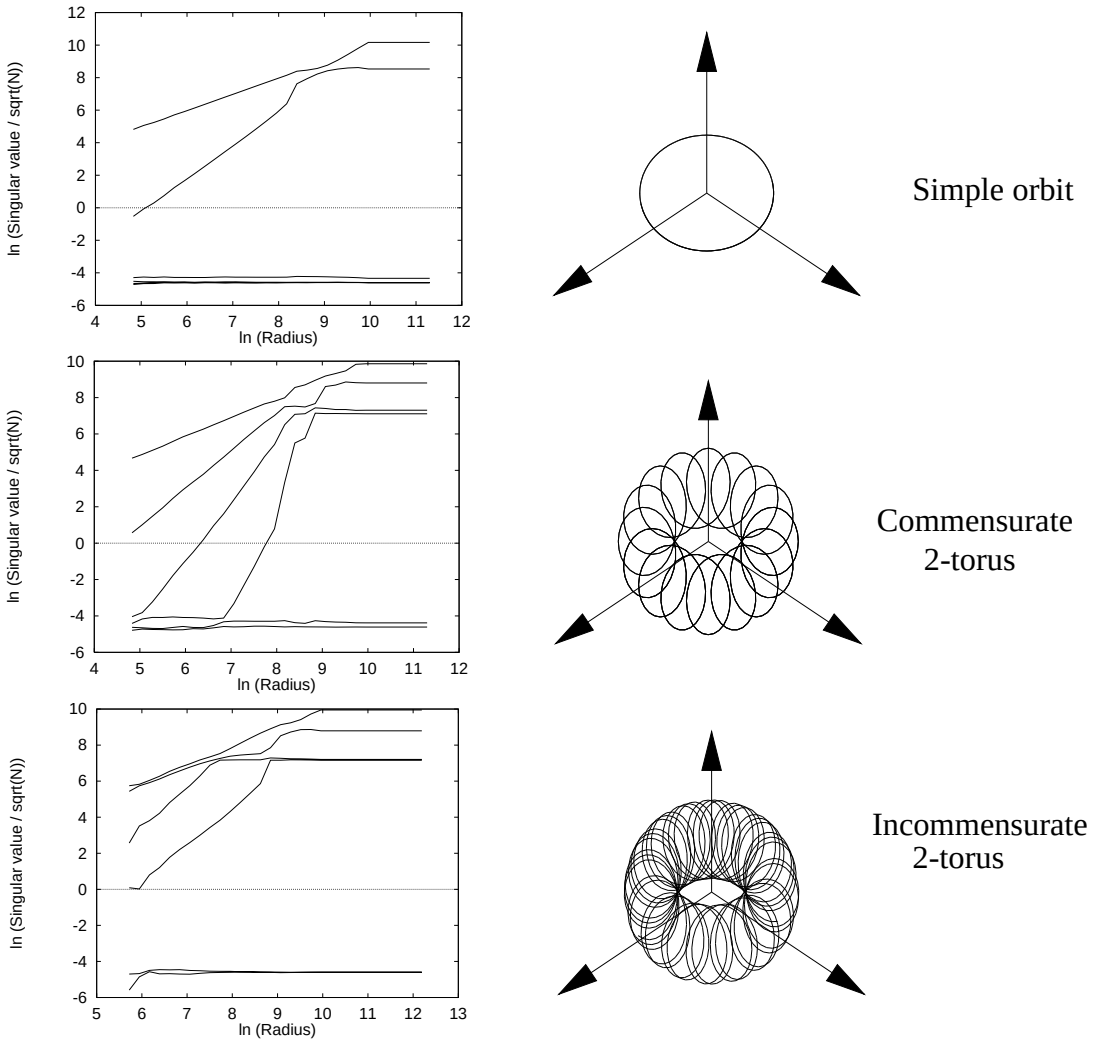


Figure 4.3: *Local Singular Value Decomposition analysis for a single frequency orbit, 2-torus and an incommensurate 2-torus with low level background noise.*

seen in the spectra for the torus where the top two values are the same as for the simple orbit, a torus is nothing more than a simple orbit with another smaller orbit added, but two new values have appeared with gradients of 3 and 4. These again signify curvature of the manifold and not a rise in the dimension; if you zoomed in on the trajectory then at a small enough scale, even for an infinite amount of data, you would only see a line therefore the trajectory is still one dimensional.

An incommensurate torus has a trajectory which does not simply repeat itself, rather it passes around and around covering the surface of the torus making a solid shape reminiscent of a doughnut. Now we have a two dimensional plane that has been curved as is seen by the addition of another value rising with gradient one. Since we see two rising values we have a system with two degrees of freedom; in this case we know that the system is of integer dimension and therefore we don't need to add one to get the dimension of the system.

A few other features of the plots should be pointed out. There is a flattening off of the values when the size of the radius exceeds the boundary of the attractor. This can even be seen to occur at different sizes according to which dimension is being measured; the incommensurate torus has one value that levels at a radius of 2000 and one at 20000, these are the diameters of the two orbits. At very low singular values it is possible to see the background noise, for these examples this is simulated as Gaussian noise with a variance of 0.01, which is seen as flat singular values at the bottom of the plots.

4.3 Lyapunov Spectra

The algorithm for extracting exponents from a time series is complex and requires care in its application and the interpretation of its results. This section describes the algorithm itself, which is broken down into a series of subsections, each dealing with a separate process the sum of which make up the extraction algorithm, and its application to chaotic time series introducing two major novel improvements that overcome the problems of short data sets and noise contamination.

4.3.1 The algorithm

This section describes the details of the algorithm used to calculate Lyapunov exponents.

Overview of the algorithm

Before dissecting the algorithm into individual sections it is useful to have some feel of the overall structure. Figure 4.4 shows the basic structure of the algorithm along with some extra annotations which show the individual stages. An attractor is formed in state space using either time delay embedding [62] or singular value decomposition [85] and the local dynamics for a series of points around that attractor are estimated via the tangent map, $\boldsymbol{T}$. $\boldsymbol{T}_i$, is calculated by examining how a local area evolves from one point to another, which is achieved by finding several nearby points and forming a neighbourhood matrix, $\boldsymbol{B}_i$, and its evolved matrix , $\boldsymbol{B}_{i+1}$. $\boldsymbol{T}$ is directly calculated from the neighbourhood matrices and then by performing a QR decomposition on the tangent map at each step the exponents can be calculated as shown in Figure 4.4.

Time Series Embedding

The time series data is embedded into state space using either time delay embedding or singular value decomposition embedding as described earlier in the chapter.

Choosing The Neighbourhood

To calculate the Lyapunov exponents of the system, an average of the local Lyapunov exponents is taken for a number of revolutions of the attractor. To achieve this, we consider the evolution of a d-dimensional hyper-sphere as it traverses the trajectory. First, we must define how the points within that d-dimensional hyper-sphere are chosen.

If the time step to be considered is sufficiently small, then we can consider the evolution of our hyper-sphere to be a linear process over that time step. However, for this to be the case, the radius of the sphere itself must be small so that it does not contain points

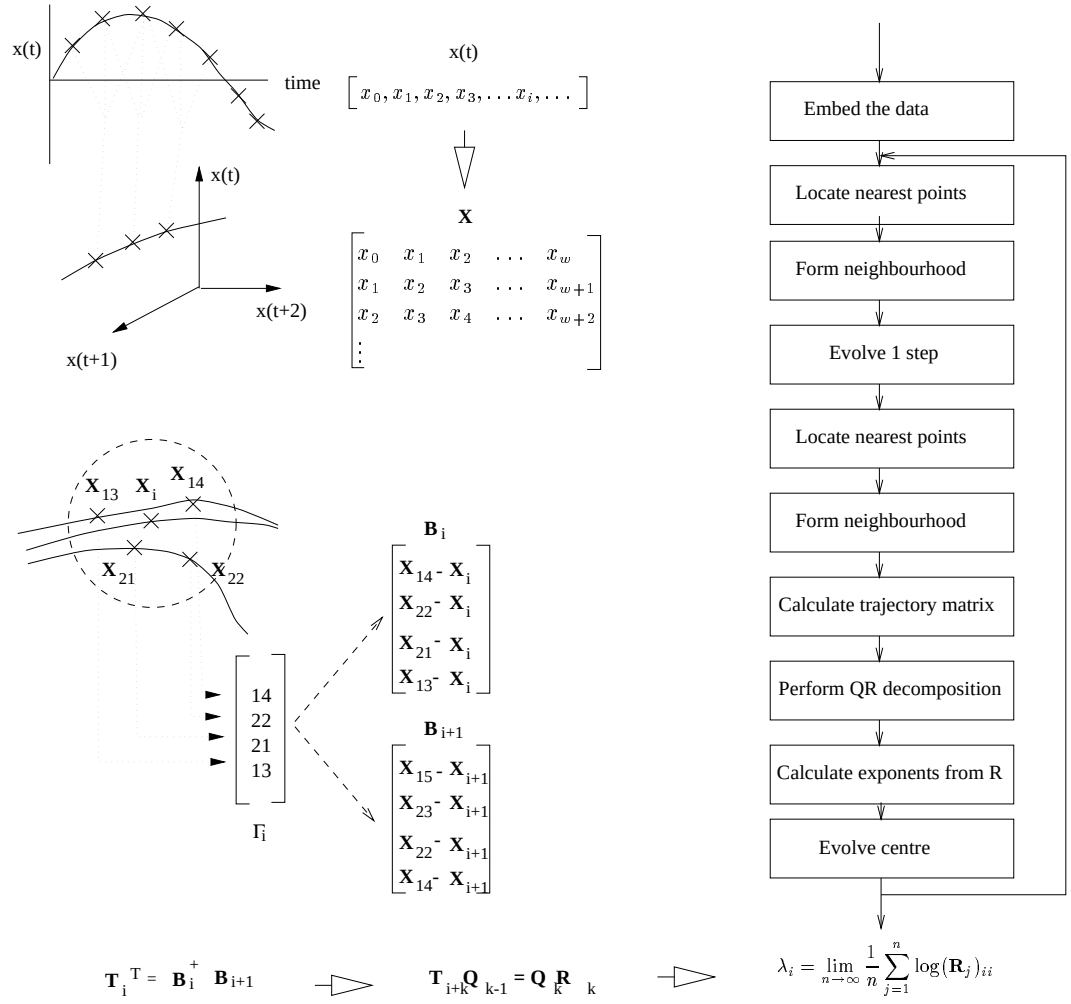


Figure 4.4: *Overview of the algorithm*

outside the local linear space. In order to achieve this, the radius ϵ is increased until sufficient points are found within the sphere, so that the minimum possible radius is used. This total number of points, M, in the hyper-sphere is one of the parameters that can be changed when applying the algorithm, and choosing a suitable value of M is addressed in the next section.

A neighbourhood set $\mathbf{\Gamma}_i$ is constructed, containing a list of the row numbers of $\hat{\boldsymbol{X}}$ corresponding to the points within the ϵ radius hyper-sphere centred on $\underline{x}_i$;

$$\mathbf{\Gamma}_i(\underline{x}_i, \epsilon) = (k \in J : \epsilon > |\, \underline{x}_k - \underline{x}_i \,|) \tag{4.6}$$

where $\underline{x}_n$ represents the nth row of matrix $\hat{\boldsymbol{X}}$ and $J = [1, 2, 3 \ldots N - a]$ such that all points will be further away from the end of the time series than the number of evolutions a to be carried out. However, not all of these points are necessarily of use to us, as some may be *false nearest neighbours* due to an inadequate embedding of the attractor causing two arms of the attractor to appear close together. Consequently, to ensure that all points are evolving along suitable trajectories, we include a check that the evolved points remain within a 2ϵ radius hypersphere. To accommodate this, the radius ϵ is increased until M points are found within the initial hyper-sphere that evolve to be within a radius 2ϵ hyper-sphere centred on the evolved point x_{i+a}. The value of 2ϵ is chosen to ensure that sufficient neighbours will be found. If this value is too large then false neighbours will get chosen and if is too small then their will not be enough neighbours that meet the criteria and so the algorithm will once again have to choose false neighbours. Our expression for the neighbourhood set now becomes

$$\mathbf{\Gamma}_i(\underline{x}_i, \epsilon) = (k \in J : \epsilon > |\, \underline{x}_k - \underline{x}_i \,|), 2\epsilon > |\, \underline{x}_{k+a} - \underline{x}_{i+a} \,|) \tag{4.7}$$

where a is the number of evolve steps to be taken before re-initialisation of the neighbourhood. $\mathbf{\Gamma}_i$ thus contains a list of the row numbers of $\hat{\boldsymbol{X}}$ corresponding to the points to be used in the local neighbourhood calculation.

The neighbourhood matrix $\boldsymbol{B}_i$ can now be constructed, containing a set of vectors within the ϵ radius hyper-sphere that will be used to calculate the local dynamics of the system.

$$\boldsymbol{B}_i = \begin{pmatrix} \underline{\gamma}_1 - \underline{x}_i \\ \underline{\gamma}_2 - \underline{x}_i \\ \vdots \\ \underline{\gamma}_b - \underline{x}_i \end{pmatrix} \tag{4.8}$$

where $\underline{\gamma}_n$ represents the row of $\hat{\boldsymbol{X}}$ indicated by the nth entry in $\boldsymbol{\Gamma}_i$.

As the neighbourhood moves around a chaotic attractor, the points within the hypersphere eventually become separated, as the non-linear aspect of long term evolution takes effect. Consequently it becomes necessary to reconstruct a new neighbourhood. As with the number of neighbours and the number of sub-groups, the number of evolve steps carried out between reinitialisations of the neighbourhood is a coefficient that can be varied in order to achieve some confidence in the results. This is discussed in a later section.

Calculating The Tangent Maps

The next stage is to estimate the *tangent map* $\boldsymbol{T}_i$ which operates on the neighbourhood matrix $\boldsymbol{B}_i$ to produce the evolved neighbourhood matrix $\boldsymbol{B}_{i+a}$. The eigenvalues of this tangent map give the local Lyapunov exponents at that point in phase space, in that $\boldsymbol{T}_i$ defines the linear transformation from $\boldsymbol{B}_i$ to $\boldsymbol{B}_{i+a}$.

The tangent map $\boldsymbol{T}_i$ for the first a evolve steps from the initial neighbourhood $\boldsymbol{B}_i$ is estimated from a series of matrix operations τ_k on each point in the neighbourhood. So

$$\underline{b}_{i+a_k}^T = \tau_k \underline{b}_{i_k}^T \tag{4.9}$$

where $\underline{b}_{i_k}$ is the kth row of $\boldsymbol{B}_i$. Combining each of these τ_k gives an expression for the transformation of the whole neighbourhood

$$\boldsymbol{B}_{i+a}^T = \boldsymbol{T}_i \boldsymbol{B}_i^T \tag{4.10}$$

which can be rewritten as

$$\boldsymbol{B}_{i+a} = \boldsymbol{B}_i \boldsymbol{T}_i^T. \tag{4.11}$$

Note that, in solving this equation, $\boldsymbol{T}_i$ is an approximation to the transformation of any particular point in the neighbourhood, based upon an assumption of a linear transformation throughout the neighbourhood. Figure 4.5 shows this graphically.

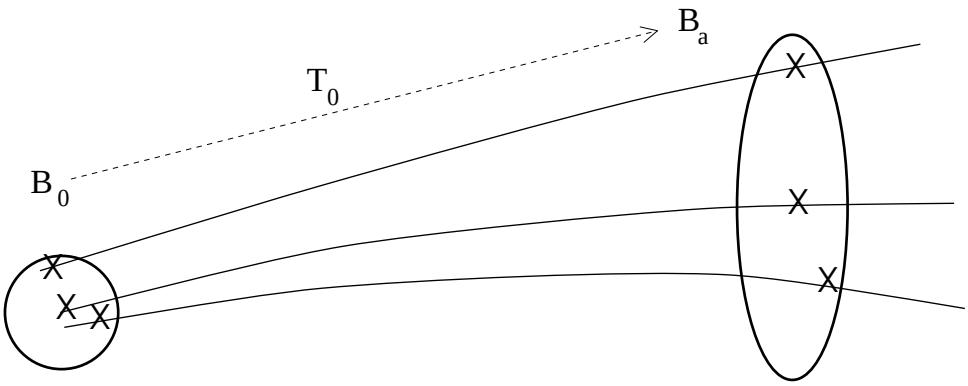


Figure 4.5: *Evolution of hypersphere for* a *time steps around an attractor.*

The evolved neighbourhood matrix $\boldsymbol{B}_{i+a}$ is constructed by taking the values of $\hat{\boldsymbol{X}}$ a rows further down from those points contained within $\boldsymbol{B}_i$. This is achieved by adding a to each of the row numbers contained in $\boldsymbol{\Gamma}_i$ and then constructing the evolved neighbourhood matrix $\boldsymbol{B}_{i+a}$ from this evolved neighbourhood set in the same manner as shown previously, ensuring that ordering of the points is maintained.

In order to calculate $\boldsymbol{T}_i$ from Equation 4.11, the inverse of $\boldsymbol{B}_i$ is needed, i.e.

$$\boldsymbol{B}_i^{-1}\boldsymbol{B}_{i+a} = \boldsymbol{T}_i^T \tag{4.12}$$

but, in general, the inverse of $\boldsymbol{B}_i$ does not exist. Moore and Penrose [108] have described the *pseudo-inverse* of a matrix, however, and this is the method adopted here. The pseudo-inverse of $\boldsymbol{B}$, represented as $\boldsymbol{B}^+$, has the following properties if $\boldsymbol{B}$ is full rank

$$\boldsymbol{B}^+\boldsymbol{B} = \boldsymbol{I} \qquad\qquad \boldsymbol{B}\boldsymbol{B}^+ \neq \boldsymbol{I}. \tag{4.13}$$

Thus Equation 4.11 can be written as

$$\boldsymbol{B}_i^+\boldsymbol{B}_{i+a} = \boldsymbol{T}_i^T \tag{4.14}$$

and the desired tangent map $\boldsymbol{T}_i$ can be estimated.

The pseudo-inverse of $\boldsymbol{B}_i$ is constructed from its singular value decomposition

$$\boldsymbol{B}_i = \boldsymbol{S}_i \Sigma_i \boldsymbol{C}_i^T \tag{4.15}$$

where $\boldsymbol{S}_i$,Σ_i and $\boldsymbol{C}_i$ are calculated as described in the subsection on time series embedding. The pseudo-inverse of $\boldsymbol{B}_i$ is defined by

$$\boldsymbol{B}_i^+ = \boldsymbol{C}_i \Sigma_i^+ \boldsymbol{S}_i^T \tag{4.16}$$

where Σ_i^+ is the pseudo-inverse of Σ_i, calculated simply by replacing each non-zero element of the diagonal matrix Σ_i with its reciprocal.

From Equation 4.12, the tangent map $\boldsymbol{T}_i$ can now be calculated, to describe the local linear dynamics at that point in the system. In order to assess the global dynamics, this process must be repeated across a sufficient expanse of the attractor and the local exponents averaged to give the global exponents.

Averaging The Exponents

We have now achieved an algorithm for calculating a string of tangent maps for a point evolving around an attractor. The final hurdle is to formulate an expression for averaging the contraction and expansion of the phase space represented therein.

The method for formulating the global Lyapunov exponents from the local mappings is taken from the work of Eckman and Ruelle [67], which is based upon the QR-factorisation technique.

In general, any matrix $\boldsymbol{A}$ can be written

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R} \tag{4.17}$$

where $\boldsymbol{Q}$ has orthogonal columns and $\boldsymbol{R}$ is a square upper-right triangular matrix with positive values on the diagonal. The method for constructing these matrices is best described in a series of steps [109]:

i) Write $\boldsymbol{A}$ as a series of columns $\boldsymbol{A} = [\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_m]$.

ii) If $\underline{a}_1 = 0$, set $\underline{q}_1 = 0$; otherwise set $\underline{q}_1 = \frac{\underline{a}_1}{\sqrt{(\underline{a}_1^* \underline{a}_1)}}$.

iii) For each $k = 2, 3, \ldots, m$

$$\underline{y}_k = \underline{a}_k - \sum_{i=1}^{k-1} (\underline{q}_i^* \underline{a}_k) \underline{q}_i .$$

iv) If $\underline{y}_k = 0$, set $\underline{q}_k = 0$; otherwise set $\underline{q}_k = \frac{\underline{y}_k}{\sqrt{(\underline{y}_k^* \underline{y}_k)}}$.

v) Build up $\boldsymbol{Q}$ from the columns $\boldsymbol{Q} = [\underline{q}_1, \underline{q}_2, \ldots, \underline{q}_m]$.

vi) $\boldsymbol{R}$ is calculated as $\boldsymbol{R} = \boldsymbol{Q}^T \boldsymbol{A}$, since $\boldsymbol{Q}$ is an orthogonal matrix.

This method is, in effect, the result of applying a Gram-Schmidt orthogonalisation to each of the columns of $\boldsymbol{A}$.

Now this QR-factorisation is applied to the problem at hand. We start with a purely arbitrary orthogonal matrix, which we label $\boldsymbol{Q}_0$ for reasons which will become apparent; the columns of this matrix we take as the basis for the initial tangent space. For simplicity's sake, we take this arbitrary matrix to be the identity matrix.

$$\boldsymbol{Q}_0 = \boldsymbol{I}_m \tag{4.18}$$

By applying our first tangent map $\boldsymbol{T}_i$ to this basis set, we construct the set of tangent vectors at the next evolved point on the trajectory, $\boldsymbol{T}_i \boldsymbol{Q}_0$. Carrying out a QR-factorisation on this matrix as in equation 4.17, produces an orthogonal set of basis vectors for the evolved tangent space, i.e.

$$\boldsymbol{T}_i \boldsymbol{Q}_0 = \boldsymbol{Q}_1 \boldsymbol{R}_1 \tag{4.19}$$

where $\boldsymbol{Q}_1$ is the new set of basis vectors. This process is repeated through the string of tangent maps, giving a string of orthogonalised basis vectors as the attractor evolves. So, in general

$$\boldsymbol{T}_{i+k} \boldsymbol{Q}_{k-1} = \boldsymbol{Q}_k \boldsymbol{R}_k . \tag{4.20}$$

Now, in order to trace the evolution of the attractor, i.e.

$$\boldsymbol{B}_{i+1} = \boldsymbol{T}_i \boldsymbol{B}_i \tag{4.21}$$

$$\boldsymbol{B}_{i+2} = \boldsymbol{T}_{i+1} \boldsymbol{B}_{i+1} = \boldsymbol{T}_{i+1} \boldsymbol{T}_i \boldsymbol{B}_i \tag{4.22}$$

the tangent maps $\boldsymbol{T}_i, \boldsymbol{T}_{i+1} \ldots$ need to be multiplied together,

$$\boldsymbol{T}_{i\to n} = \boldsymbol{T}_n \boldsymbol{T}_{n-1} \ldots \boldsymbol{T}_{i+1} \boldsymbol{T}_i \tag{4.23}$$

but by rewriting equation 4.20 as

$$\boldsymbol{T}_{i+k} = \boldsymbol{Q}_k \boldsymbol{R}_k \boldsymbol{Q}_{k-1}^{\mathsf{T}} \tag{4.24}$$

and substituting this into equation 4.23

$$\boldsymbol{T}_{i\to n} = \boldsymbol{Q}_n \boldsymbol{R}_n \boldsymbol{Q}_{n-1}^{\mathsf{T}} \boldsymbol{Q}_{n-1} \boldsymbol{R}_{n-1} \boldsymbol{Q}_{n-2}^{\mathsf{T}} \ldots \boldsymbol{Q}_2 \boldsymbol{R}_2 \boldsymbol{Q}_1^{\mathsf{T}} \boldsymbol{Q}_1 \boldsymbol{R}_1 \boldsymbol{Q}_0^{\mathsf{T}} \tag{4.25}$$

the expression can be written

$$\boldsymbol{T}_{i\to n} = \boldsymbol{Q}_n \boldsymbol{R}_n \boldsymbol{R}_{n-1} \ldots \boldsymbol{R}_2 \boldsymbol{R}_1 \tag{4.26}$$

since the $\boldsymbol{Q}$ matrices are orthogonal and $\boldsymbol{Q}_0$ has been set to the identity matrix. It is evident that the necessary information can be taken directly from the $\boldsymbol{R}$ matrices as they are calculated. In fact the diagonal entries of the $\boldsymbol{R}$ matrices contain the relationship between these bases and the evolution of the $\boldsymbol{Q}$ matrices due to the tangent maps. This means that these diagonal values effectively contain the local Lyapunov exponents, so the global exponents λ_i can be calculated from

$$\lambda_i = \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} \log(\boldsymbol{R}_j)_{ii} \tag{4.27}$$

where n is the number of evolve steps carried out.

4.3.2 Using the algorithm

The previous section described in detail our algorithm for computing the Lyapunov exponents of a system from a single time series, however, due to the complexity of the algorithm, this is not the whole story. Care must be taken both in the choice of parameters used when applying the algorithm to a particular time series, and in the interpretation of the results derived from it. This section is divided into an explanation of the parameters that can be varied and then an example of its application to a Lorenz time series.

The Parameters

It has been intimated that there are a number of parameters to be chosen when applying the Lyapunov exponent algorithm to a time series. In fact, the best approach is to vary each of these parameters across a suitable range of values and then to assimilate the results in some way. The main parameters of interest are:

- *number of neighbours.* The number of neighbours used can be critical to the accuracy of the calculation. Increasing the number of neighbours ought to increase the accuracy of the calculation, particularly with respect to averaging out noise, but it also necessarily increases the radius of the neighbourhood. If this radius grows too large, then the linearity of the neighbourhood becomes compromised and the locally linear assumption is no longer valid. Equally there must be a sufficient number of points so that the $\boldsymbol{B}$ matrix fully represents the attractor's dynamics. A figure of *at least* $2\mathtt{d}+1$, where $\mathtt{d}$ is the dimension of the system, has been recommended for the number of vectors in $\boldsymbol{B}$ [90], although if the data has any noisy corruption then it is advantageous to increase this number ten or twenty fold. This sort of increase can only be achieved if the data records are sufficiently long which is a problem that is addressed later in this chapter.

- *size of SVD window.* In the presence of noise, a larger window is recommended, as it reduces the noise content of the embedded attractor. Indeed, since the *SVD* is used for a global embedding, it is advisable to use a window length approximately equal to a single revolution of the attractor. However, it should be noted that a longer window appears to degrade the calculation of the negative exponent [29].

- *dimension.* The problem of choosing the correct dimension to embed a time series in is a thorny one. Indeed, it is a problem to which an adequate solution has not really been found. Various methods have been suggested, from a range of fractal dimension measures [80] to examining the *SVD* spectrum [85], but, in practice, the results provided by these algorithms appear to be ambiguous when applied to a measured noisy time series. A full investigation of this question is beyond the scope of this chapter, and is an ongoing research topic within the field. However, by using the algorithm at a range of different embedding dimensions, given approximately by the dimension measures, it usually becomes clear what the most appropriate dimension is since the results become more

stable to parameter changes. A further problem is the constraint placed on the embedding by Takens' theorem; that the system must be embedded into a $m \geq 2d + 1$ state space. This has the consequence that a fully embedded system has $d + 1$ spurious exponents. Taken's theorem gives a theoretical sufficient bound but many systems in practice can be embedded in much lower spaces. The systems used in this thesis seem to embed sufficiently into d dimensional space and therefore the spurious exponents do not arise, however in general this may not be the case and as such is a real problem that needs to be addressed. One solution is suggested in a paper by Darbyshire and Broomhead [29] and a full description of this technique here would cloud the overall algorithm description. In broad terms it is suggested that the system can be embedded into m dimensions initially and that the neighbourhoods should be chosen in that m dimensional space. Once the neighbourhood matrices have been defined for a point, then it is possible to perform another singular value decomposition, as described earlier, to reduce the neighbourhood matrices from m to d dimensions. The rest of the calculation is performed as before in the d dimensional space therefore producing the correct number of exponents. The interested reader is directed to the original paper [29] for full implementation details.

- *evolve steps between re-initialisation.* The value a must be chosen so that a re-initialisation occurs before the points in the neighbourhood set become so widely dispersed that a hyper-sphere large enough to contain them is larger than the locally linear space. However, it is worth postponing the re-initialisation as long as possible in order to see a greater expansion or contraction over the evolve period and to minimise the effect of noise as a ratio to such expansion. Furthermore, choosing a larger evolve period increases the reliability of our false nearest neighbours test.

- *total number of evolve steps to take, i.e. the total number of tangent maps to calculate.* Obviously the higher the better, though this increases computing time. In practice, the values of the Lyapunov exponents tend to their final values asymptotically and it is easy to tell whether the algorithm has run for enough steps. It should be ensured that the neighbourhood has been evolved *at least* once around the whole attractor.

- *total number of points on the attractor.* Again, the more the better since it increases knowledge of the attractor, though obvious limits occur on storage space and computation time. Enough points are needed for a good representation of the

attractor as a whole.

It can be seen that applying the Lyapunov exponent algorithm to a time series is not an exercise to be taken lightly. There really is no substitute for becoming intimately familiar with the attractor in question and gaining a good knowledge of how the algorithm behaves under a large range of parameters for the particular time series under study.

Application to the Lorenz time series

It is instructive to show typical results for a time series for which the exponents can be directly calculated from the differential equations. The signal under consideration is generated from the Lorenz system [110]

$$\dot{X} = \sigma(Y - X) \qquad \dot{Y} = rX - Y - XZ \qquad \dot{Z} = -bZ + XY \tag{4.28}$$

with the parameters $\sigma = 16.0$, $r = 40.0$ and $b = 4.0$. Figure 4.6 shows the attractor under consideration. It is generally accepted that this attractor should be embedded in three dimensions, and has three Lyapunov exponents of size $+1.37$, 0 and -22.37 calculated by the algorithm in [84]. The time series comes from the X component.

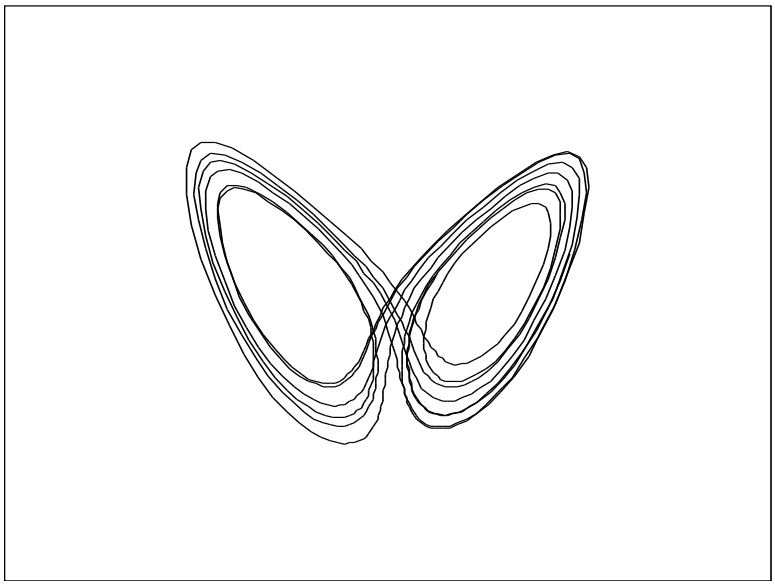


Figure 4.6: *The Lorenz attractor*

The first parameter that can be varied is the number of vectors b in the $\boldsymbol{B}$ matrix, as shown in Figure 4.7.

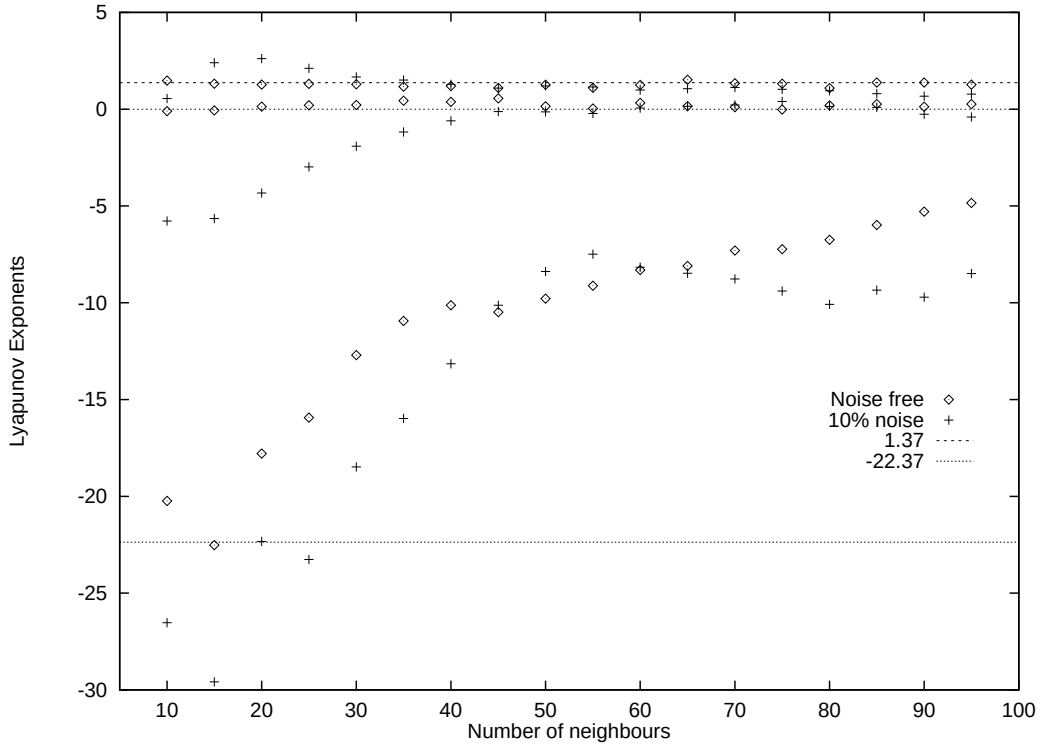


Figure 4.7: *Lyapunov exponents of Lorenz data for varying number of vectors in neighbourhood matrix. Parameters are* 49000 *points sampled at* 0.01*s;* 3000 *iterations of* 5 *evolve steps each; radius of neighbourhood* 1.0*; SVD window size of* 15 *(noise free) and* 50 *(noisy).*

The accepted values are marked on the plot and it can be seen that for the noise free case a small number of neighbours is optimal with the accuracy being compromised when too many are used. The plot also shows the results for data which has additive white Gaussian noise at 10% of the variance of the signal. It can be seen that the number of neighbours needs to be increased in order to allow the least squares technique to perform some averaging of the noise. It should be noted here that the SVD window has been increased for the noise case which has had the consequence of reducing the amplitude of the negative exponent. This is a common feature of algorithms of this nature [29] [90] and, in terms of the usefulness of the results, a reasonable approximation is attained.

One of the ways in which the algorithm can be made to account for noise is by increasing the size of the *SVD* window. Figure 4.8 shows the exponents calculated for both noisy Lorenz data and clean Lorenz data over a range of window sizes. For the clean data only 15 neighbours are used but for the noise contaminated data this number is raised to 50, with the other parameters remaining the same as for Figure 4.7. As can be seen, the introduction of noise has affected the calculation of all three

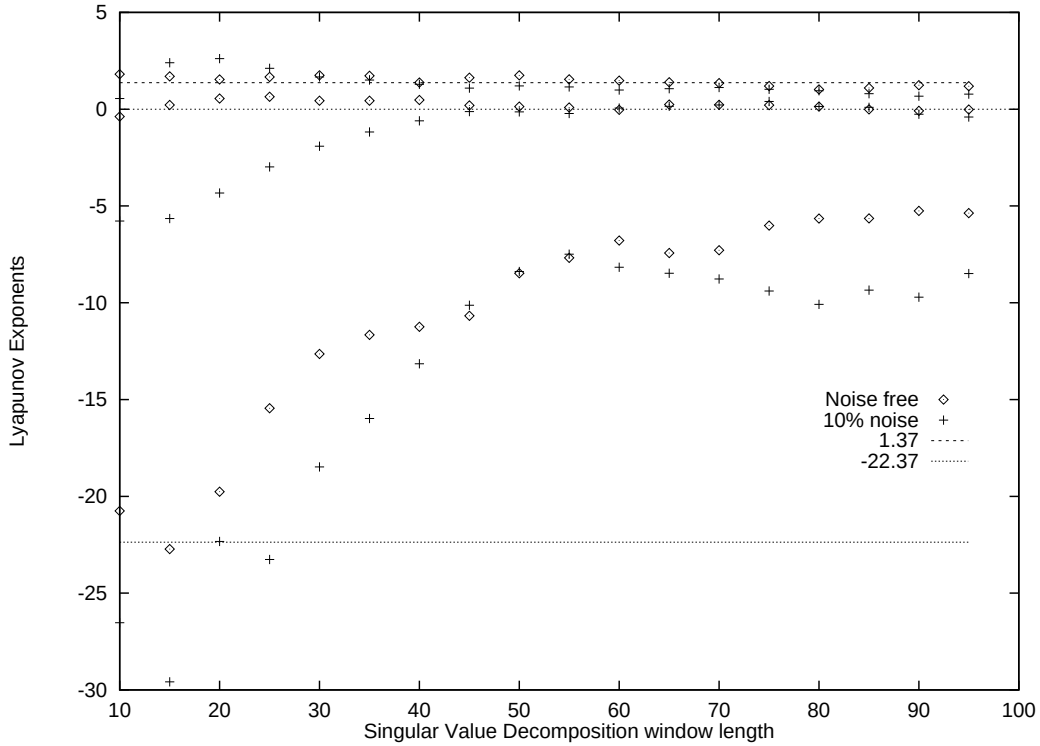


Figure 4.8: *Lyapunov exponents of Lorenz data for varying size of SVD window. Parameters are* 49000 *points sampled at* 0.01*s;* 3000 *iterations of* 5 *evolve steps each; radius of neighbourhood* 1.0*;* 15 *neighbours (noise free) and* 50 *(noisy).*

of the exponents, especially at low window sizes. Since the data under analysis is a flow, it must have a zero exponent and the plot reveals that the best zero is produced for a *SVD* window size of around 40 to 45. It is also apparent that the values level out above a window size of around 50; this value coincides with the number of samples in a typical revolution of the attractor, and therefore *SVD* window sizes above this value do not significantly increase the global information attained. It can be seen that the amplitude of the negative exponent has been greatly reduced; again this is a common problem [29] [90], but it does not necessarily affect the useful information that can be derived from such an analysis. It should again be noted that the value used for noisy data is not the optimal value for clean data. In the case of the clean data the SVD window should be about 15 in order to provide a full embedding which does not adversely affect the negative exponent.

The remaining parameter to investigate is the number of evolve steps between re-initialisations a. The results are shown in Figure 4.9 with the same parameters as the previous plot. At low values of a, the noise has a greater effect, since it represents a higher percentage of the change between consecutive points used in the calculation. As the evolve step size grows, the accuracy of the calculation improves, evinced by

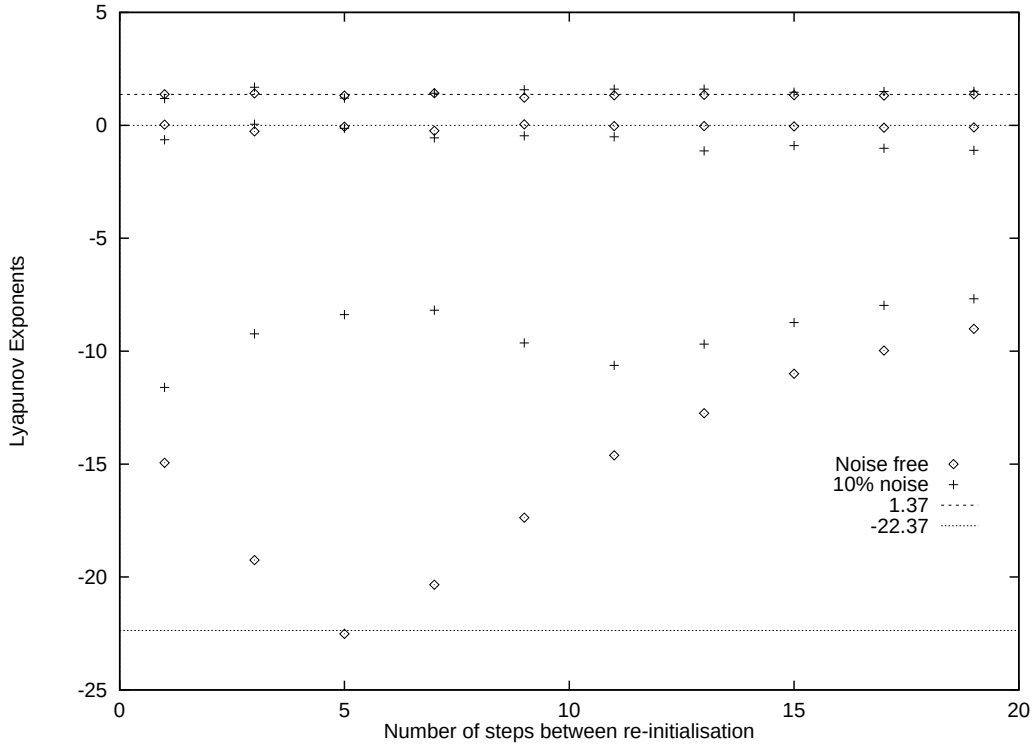


Figure 4.9: *Lyapunov exponents of Lorenz time series for varying number of evolve steps between re-initialisations* a. *Parameters are* 49000 *points sampled at* 0.01*s;* 3000 *iterations; radius of neighbourhood* 1.0; 15 *neighbours (noise free) and* 50 *(noisy); SVD window size of* 15 *(noise free) and* 50 *(noisy).*

the zero exponent becoming more clear. Again too high a value causes the assumption of a linear transformation $\boldsymbol{T}$ to lose its validity. In this case it seems that the optimal value is 5 for both clean and noisy data.

It has been noted that throughout the results presented for the noisy signal, the amplitude of the negative exponent has been grossly under-estimated. In practice, this is not the problem that it may at first appear to be. The usefulness of the Lyapunov spectrum must be borne in mind here. The Lyapunov exponents of a system essentially provide two services: first, they can be used to qualitatively label a system as chaotic or non-chaotic, in this case our noisy Lorenz time series is clearly derived from a chaotic system since we reliably find a positive and a negative exponent; secondly, they act as a quantitative comparison between two or more time series, such a comparison is only valid if these time series have been measured in a similar way and, therefore, have a similar amount of noise on them. In that case, the calculation of the negative exponents will be affected equally by the SVD noise reduction and a meaningful comparison can be achieved.

It may also be apparent that throughout the presentation of the results for the noisy

data, the zero Lyapunov exponent has been used as the yard-stick by which to optimise the parameters. This seems sensible since it should be clear whether the signal is a flow or a mapping and therefore whether to expect the existence of a zero. This is the only real *a priori* knowledge needed by our algorithm in order to gain unambiguous and accurate results.

4.3.3 Real world problems

In most real world applications the signal may suffer from two main problems; noise contamination and insufficient record length [111]. This section examines how the results from the algorithm degrade against these parameters and show a novel approach to reducing the effect of small record length.

Noise and data length

The presence of additive noise on any signal causes great problems to the algorithm since both the accuracy of the trajectory matrix is affected and the chance of picking a false neighbour is increased. Figure 4.10 shows how the algorithm performs when the signal is corrupted by additive Gaussian noise of levels up to 30% of the signal variance (-4.7dB signal to noise ratio).The algorithm performs well for low levels of noise but the exponents begin to fall off at noise levels greater than about 15% of the signal.

A short record length, the number of points in the time series, causes the algorithm to use too large a radius in its search for near neighbours since the attractor manifold is not sufficiently covered by the available data. The problems caused by this are twofold: an enlarged neighbourhood increases the chances of false neighbours and also reduces the validity of the linearity assumptions. The effects of short data record lengths are shown quite clearly in Figure 4.11 which gives typical results for a signal to noise ratio of 0.1 (a noise level of 10% of the signal variance).

Noise robust extraction technique

The common approach for extracting Lyapunov exponents, as used by a variety of authors [49, 85, 86, 90, 91, 112–114], is described by Broomhead and Darbyshire [29]

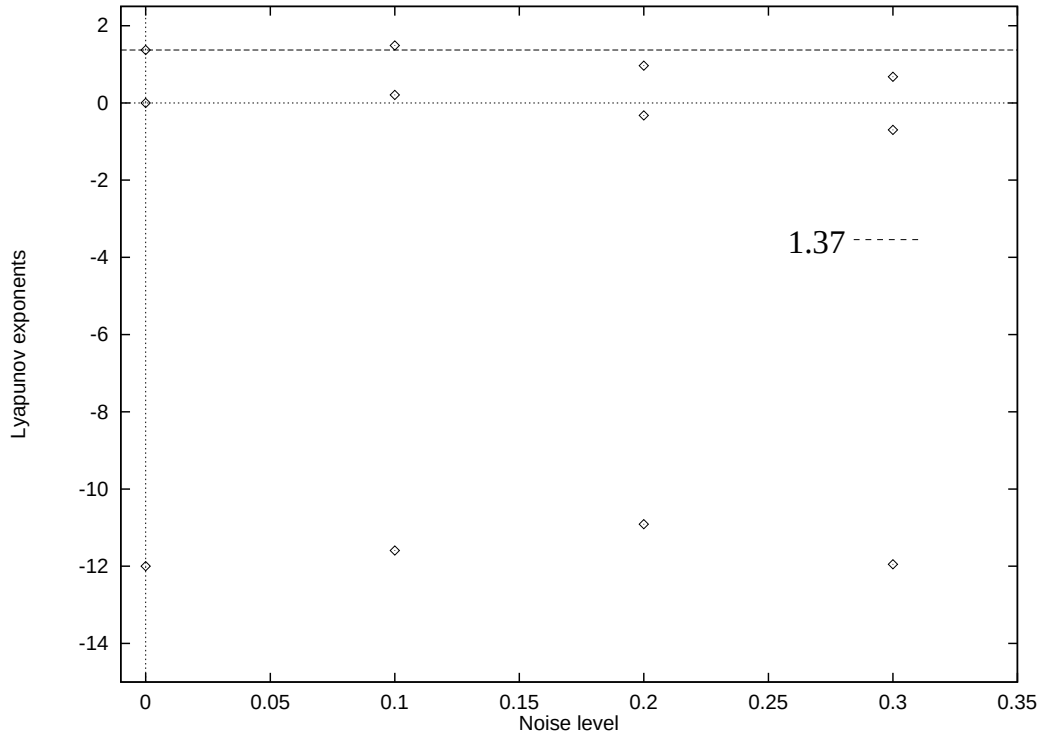


Figure 4.10: *Lyapunov exponents calculated by local SVD method for Lorenz time series with increasing additive noise. Abscissa shows noise as fraction of the variance of the signal.*

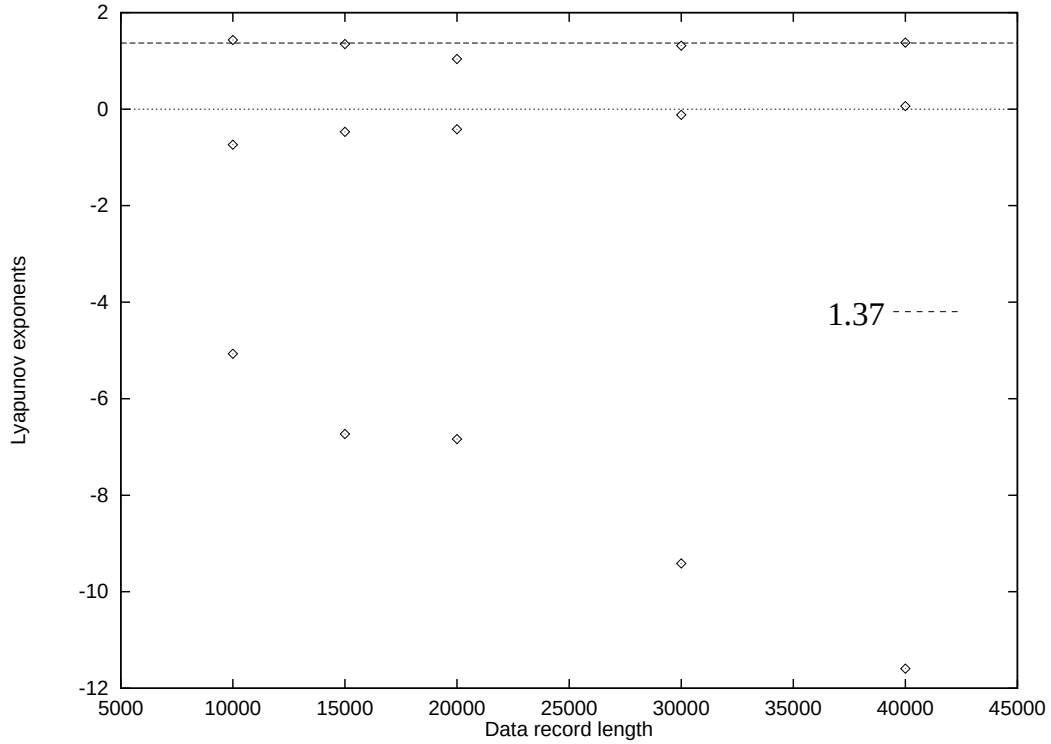


Figure 4.11: *Effects of data record length on the estimation of Lyapunov exponents.*

which, in the absence of noise, has been found to perform extremely well. However, the algorithm's robustness to noise is not sufficient to enable the algorithm to be used in many real world situations where noise corruption is a factor. In the conventional algorithm, see [29], a neighbourhood matrix $\boldsymbol{B}_i$ is constructed to contain a set of vectors within the ϵ radius hyper-sphere that is used to calculate the local dynamics of the system. In this section a novel method for defining the neighbourhood matrix to enhance the algorithm's performance in noise is presented. To provide fair comparison results are included using all the noise robustness measures suggested by Darbyshire to provide a milestone by which our algorithm can be evaluated.

In our algorithm M neighbours are located and placed in the neighbourhood set $\boldsymbol{\Gamma}_i$. These points are divided into b sub-groups which are averaged in the $\boldsymbol{B}_i$ matrix, each sub-group resulting in a single horizontal vector of $\boldsymbol{B}_i$. A full average would include a division by the number of elements but this is redundant in this calculation and therefore is not performed. Consequently the $\boldsymbol{B}_i$ matrix is formed by

$$\boldsymbol{B}_i = \begin{pmatrix} \underline{\gamma}_1 + \underline{\gamma}_{b+1} + \dots \\ \underline{\gamma}_2 + \underline{\gamma}_{b+2} + \dots \\ \underline{\gamma}_3 + \underline{\gamma}_{b+3} + \dots \\ \vdots \\ \underline{\gamma}_b + \underline{\gamma}_{2b} + \dots \end{pmatrix} \tag{4.29}$$

where $\underline{\gamma}_n$ represents the row of $\hat{\boldsymbol{X}}$ indicated by the nth entry in $\boldsymbol{\Gamma}_i$.

Creating a neighbourhood matrix in this way improves the noise performance of our algorithm in comparison to the conventional approach. The reason for this improvement is that an average of M/b vectors is used for each entry in the neighbourhood matrix. This averaging of a number of vectors will tend to cancel out the noise. It should be pointed out that the least squares solving of the matrix that is included in the conventional approach [29] does also 'average' the vectors but it would seem that by making the averaging explicit, improved noise robustness can be achieved.

As an example of the algorithm's performance Figure 4.12 shows how increasing the number of subgroups improves the exponent estimation. The corresponding estimates taken using the same number of neighbours without averaging is also shown for comparison. The data under examination is Lorenz data with exponents of 1.37, 0

and -22.37. The data has been corrupted by Gaussian noise with a variance of 20% of the data. The parameters used are those given in the paper by Darbyshire and Broomhead[2] [29] and have been chosen to provide optimal noise robustness.

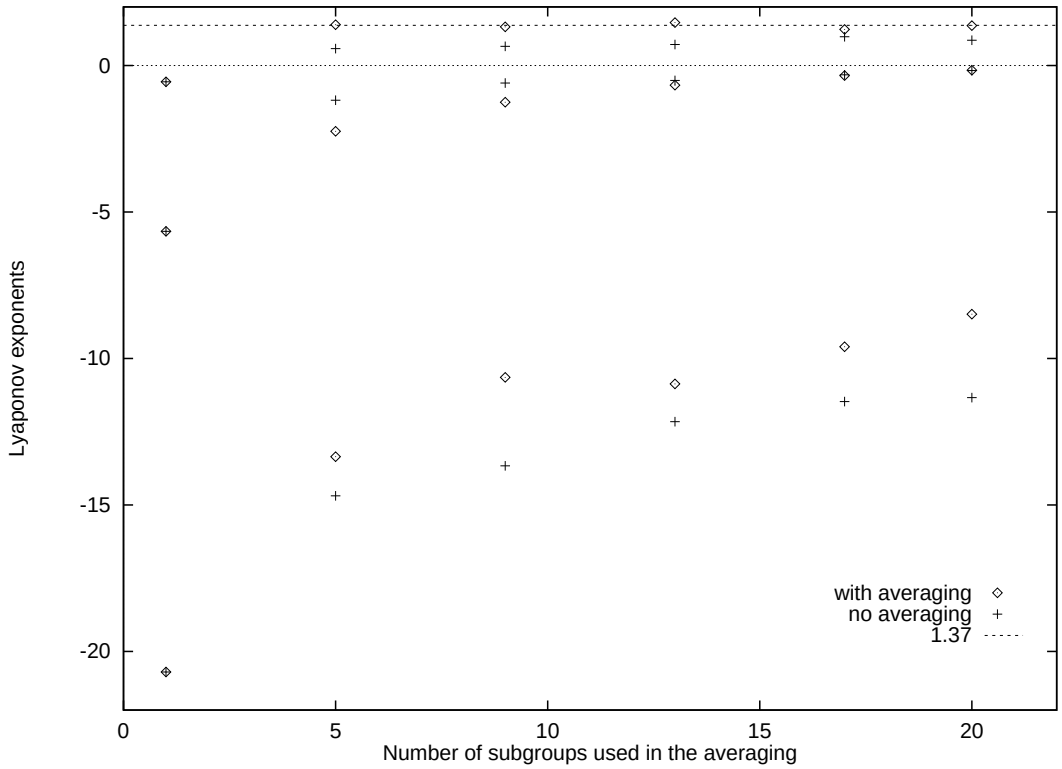


Figure 4.12: *Lyapunov exponents of noisy Lorenz time series for varying number of averages used in the neighbourhood matrix* $\boldsymbol{B}$*. Parameters are* 49000 *points sampled at* 0.01s*; global embedding dimension* 7*; local embedding dimension* 3*; annular neighbourhood;* 3000 *iterations of* 4 *evolve steps each;* 20 *vectors in* $\boldsymbol{B}$*; SVD window size of* 50*.*

The figure clearly shows a vast improvement in the accuracy of the zero and the positive exponent whilst the negative exponent is shown to be be underestimated. It is a common problem that noise robustness is achieved at the cost of the negative exponent [29] and thus a slight further degradation is not a real problem.

The algorithm's robustness to noise is shown in Figure 4.13 which shows how the accuracy of the algorithm degrades as the noise level is increased. Noise level is given as the fraction of noise variance to signal variance. Again to allow fair comparison results for no averaging but the same numbers of neighbours, are included showing that our algorithm affords a definite improvement.

[2] the Darbyshire and Broomhead approach uses an annular neighbourhood rather than a solid hypersphere

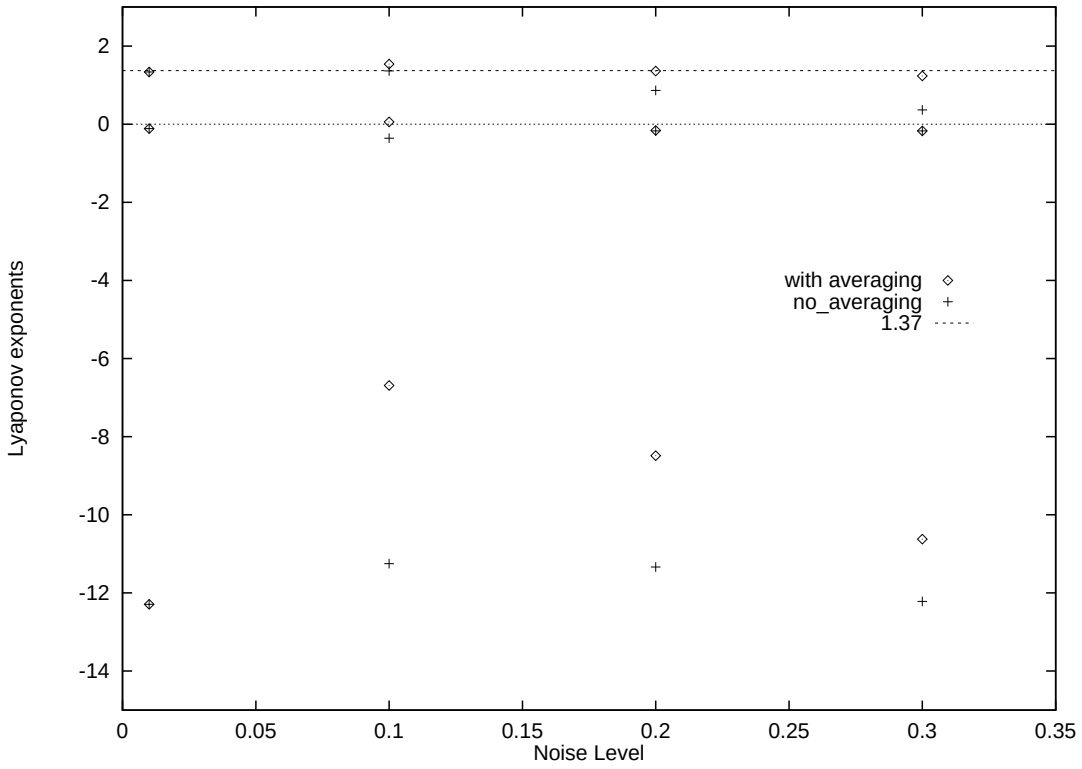


Figure 4.13: *Lyapunov exponents calculated by both the conventional technique and the averaging method for the Lorenz time series with increasing additive noise. Abscissa shows noise as fraction of the variance of the signal. Parameters used are as before.*

Data concatenation

In many real world systems the data records that can be collected will only be available for short bursts forcing us to use small data records for the analysis. This section shows a novel technique that allows the use of multiple records from the same system to be 'joined' so that the effect of one long record is achieved.

Assuming that a number of records can be obtained from the system where the underlying attractor for each record is the same, i.e. the system remains stationary from record to record, then a composite attractor can be built up as shown in Figure 4.14.

Each record is embedded as it would be normally and then sequentially joined to form the $\boldsymbol{X}$ matrix. At the same time a new vector, $\boldsymbol{X}\_\mathtt{check}_i$, must also be constructed which holds a 1 or a 0 according to whether the point is an end point of a section. This is a flag notifying that the point can not be used since it can not be evolved forwards in time. It is now a simple task to include a check in the algorithm so that when it locates the nearest neighbours it also checks to see that $\boldsymbol{X}\_\mathtt{check}_i = 1$ before accepting the point. The results of a data file built up from sections taken from a Lorenz time

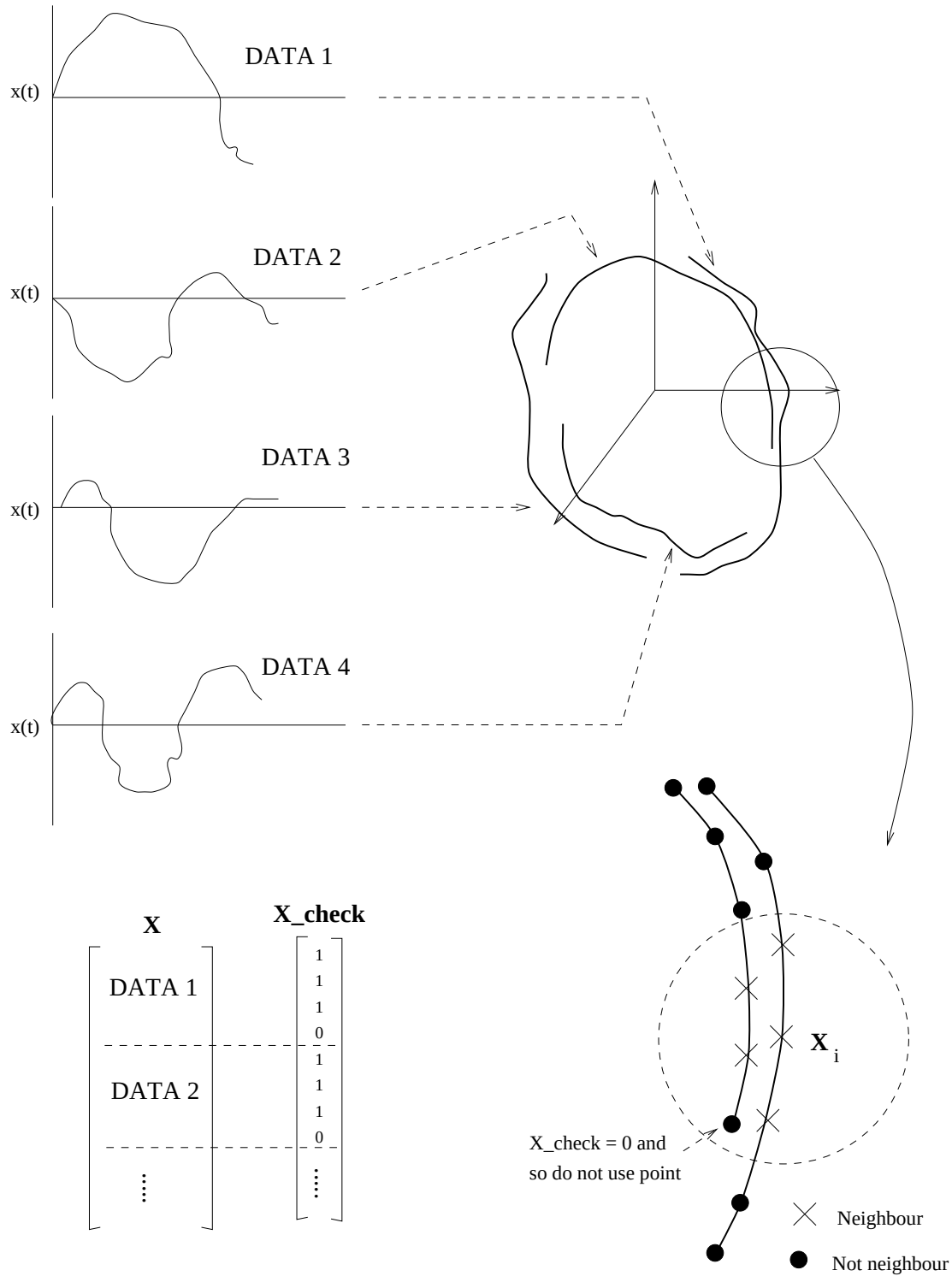


Figure 4.14: *Building up a composite attractor using multiple records*

series are shown in Figure 4.15, revealing that even though the individual sections are small the results are still as accurate as for a single long time series. In this case the attractor has been embedded using time delay embedding but the same principle can be applied to SVD embedding.

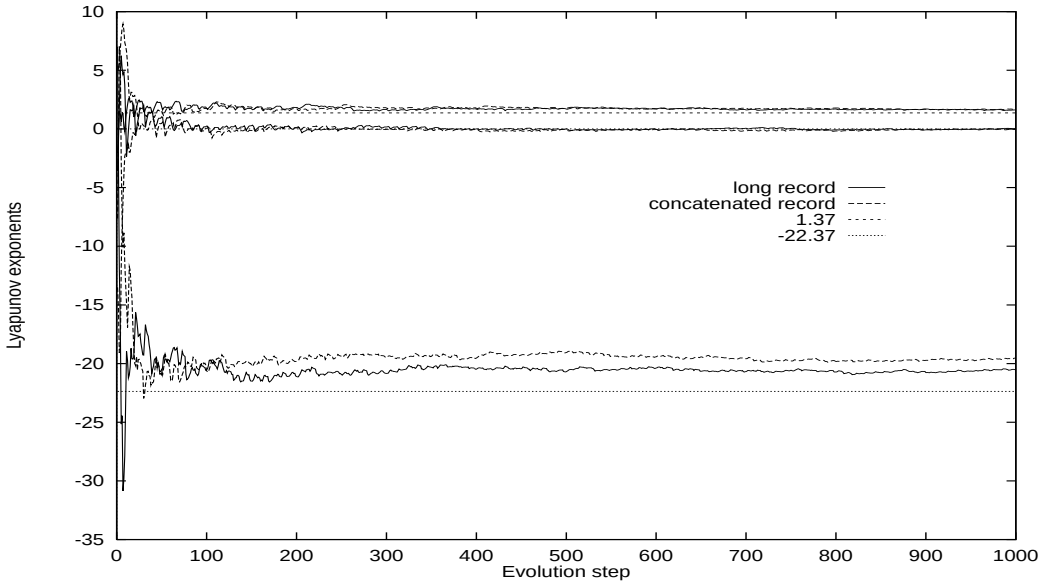


Figure 4.15: *Lyapunov exponents estimates for Lorenz data taken from a long data record (40000 points) and a composite record ($5 * 6000$ points) using time delay embedding.*

When building up the composite attractor it is important to have at least one section which is as long as the number of evolution steps that are required for convergence. This is because the algorithm must not be allowed to move from one section to another during the evolution due to the error that this would produce in the QR normalisation. This also means that to make the coding simpler for the algorithm it is best to order the concatenated segments into order of length such that the longest section comes at the start. It should also be stressed again that this technique can only be applied if the system, from which the records are taken, remains stationary.

4.4 Short Term Predictability

The short term prediction properties of the attractor can be examined using a technique suggested by Casdagli [83,115], to show whether the general behaviour of the system is best modelled by a linear stochastic or nonlinear deterministic process and furthermore whether the system is high or low dimensional. A full description of the technique

can be found in [83]. In essence the measure calculates the short term prediction error produced from a locally linear model constructed using the k nearest neighbours in state space. For low values of k the model is based on very local properties and therefore can approximate a nonlinear system very well. As the value of k is increased, the model is effectively altered from an approximation of a nonlinear deterministic model to a linear stochastic model. By examining the way in which the prediction error changes over the range of k it is possible to determine whether the system is linear or nonlinear. A basic desciption of the algorithm is as follows.

- Divide the data set into two sections : a fitting set $x_1, \ldots, x_{N_f}$ and a test set $x_{N_f+1}, \ldots, x_N$.
- Embed to m dimensions using time delay embedding.
- Choose at random a test vector, $\underline{x}_i$, from the test set.
- Compute and order the distances d_{ij} from $\underline{x}_i$ to vector $\underline{x}_j$ in the fitting set.
- Taking the k nearest neighbours from the neighbourhood matrix $\boldsymbol{B}_i$ and the corresponding evolved matrix t steps later , $\boldsymbol{B}_{i+t}$.
- Calculate an estimate of the local linear model by solving $\boldsymbol{B}_{i+t}^T = \boldsymbol{T}\boldsymbol{B}_i^T$ using the pseudo inverse method.
- Calculate the predicted value using $\underline{p}_{i+t}^T = \boldsymbol{T}\underline{x}_i^T$ and compute the error $e_i(k) = |\underline{p}_{i+t} - \underline{x}_{i+t}|$.
- Repeat for several values of i to provide an average such that the normalised geometric mean error can be calculated,

$$E_m(k) = 10^{\left(\sum_i \log e_i(k)\right)} \Big/ \sigma$$

  where σ is the standard deviation of the time series.

Results for a number of known chaotic systems, Lorenz, Henon and Mackey Glass, are summarised in Figure 4.16 which shows how the gradient varies according to the system under test.

These nonlinear systems have a clearly rising prediction error as k increases: the larger k becomes, the more seriously the nonlinearity affects the accuracy of the locally linear

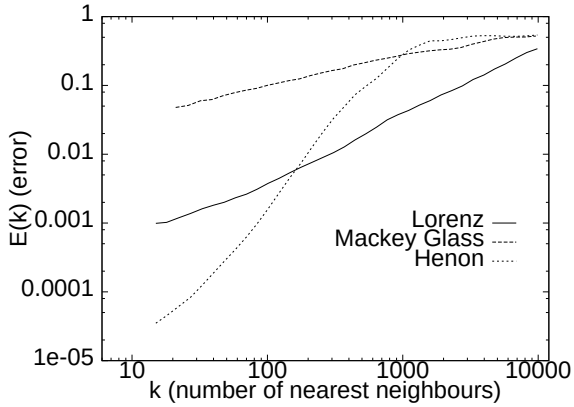


Figure 4.16: *Short term prediction error summary for chaotic systems.*

model. It can be shown [83] that the gradient of the curve can be approximated to the information dimension, d_i of the system through equation 4.30 where $E(k)$ is the geometric mean of the prediction error. This gives us a first estimate of the most suitable embedding dimension for the system.

$$\text{gradient} = \log E(k) / \log k = 2/d_i. \tag{4.30}$$

It should be noted that the information dimension, in this chapter, is being used to ascertain the dimension of the system; the actual number of dimensions that the underlying system has, not the number of dimensions in which it should be embedded to ensure no self-crossing. If a system has an information dimension that is not an integer value then the dimension of the system should be equal to the next highest integer, for instance the Lorenz system has $d_i = 2.06$ which means a dimension of 3.

Casdagli's approach has been extended by Sugihara [116] who suggested that a nonlinear weighting term should be added such that the neighbourhood is not described in such a harsh local manner. Using such a technique with nonlinear weighting term set to 1 simply results in the Casdagli model, other values allow for particular nonlinear models to be investigated. For the purposes of investigating whether a system is linear or nonlinear then Casdagli's approach provides a simple and effective tool whilst Sugihara's technique seems to cloud the issue by adding an extra nonlinear variable which is not necessary. It should be pointed out that Sugihara's work is not aimed simply at looking at linearity and does, when applied correctly, provide additional information which may be of use in an analysis.

4.5 Summary

This chapter has presented a number of algorithms and tools for the analysis of nonlinear time series data along with two very important novel modifications to the Lyapunov spectrum calculation technique which allow for noise and short data sets. The techniques looked at in detail are time series embedding using the method of delays and SVD, mutual information, dimension measures including correlation dimension, SVD and local SVD, Lyapunov spectra and short term prediction. Particular reference has been made, by way of examples, to the fact that the effects of noise should be taken in to account when performing any such analysis and the techniques put forward have been shown to be considerably more noise resilient than conventional approaches and provide accurate indicators of the system's invariant properties.

Chapter 5
APPLICATION TO SPEECH

This chapter is organised into an introduction, a section detailing the data set and the collection technique utilised, followed by a number of sections that explore some relevant invariant geometrical properties of nonlinear dynamical systems: the embedding of the system into state space; the short term predictive properties, showing how these relate to the dimension of the attractor; the local singular value decomposition; calculation of the Lyapunov spectra using parameters taken from the preceding results. Finally some conclusions are drawn regarding the implications of this work.

5.1 Introduction

With the emergence of nonlinear dynamical analysis many researchers have striven to find out whether speech can be approximated by a nonlinear, low dimensional model [25,26,28,83,117]. These investigations have been, on the whole, inconclusive and in this chapter a full analysis is presented using a range of invariant measures which show clearly the low dimensional, nonlinear behaviour of individual vowel sounds.

Time series data provide a one dimensional projection of a system's underlying geometrical attractor which can be re-projected back into a higher dimensional state space by means of time delay embedding. If the embedding dimension is high enough then the embedded attractor should have invariant qualities that can be quantified and analysed to reveal its structure [62]. In the chapter use is made of the short term predictability [83] measure to show that the system is low dimensional and nonlinear; the correlation dimension [95] and local singular value decomposition analysis to measure the space filling qualities of the attractor; and the Lyapunov spectrum to measure the attractor's sensitivity to initial conditions [85] and to provide a qualitative feel for the dynamics of the underlying system.

Previous studies [83] that have attempted to investigate the nonlinear behaviour of speech have used simple phrases as test data sets. Unfortunately connected speech,

such as a simple phrase, is inherently non-stationary since the articulatory apparatus is continually changing position in order to generate the next sound. From a dynamical system analysis point of view such a stream of data is wholly unsuitable since the dynamics themselves are undergoing continual change and must therefore be non-stationary. A better analysis can be achieved by focusing the analysis on individual phonemes, single unambiguous sounds that form the building blocks of any language, allowing the individual dynamics of each phoneme to be investigated before attempting to generalise the complete system. This analysis takes elongated vowels from a variety of different speakers and shows that irrespective of speaker the system is nonlinear, non-chaotic and has a low dimension of the order of three or four.

5.2 The data set

Before discussing how the speech was collected it is important to underline the basic philosophy behind the choice of data set. Chaotic analysis requires large, stationary data sets in order to produce reliable results. Although a certain amount can be done to reduce these requirements, as discussed in chapter 4, they must still form the basis of any data collection philosophy. For connected speech the length of time which a speaker typically sustains an individual vowel is extremely short and subject to co-articulation effects. Performing an accurate analysis of such small segments of data, which have dubious stationarity properties, is highly impractical and therefore one has to compromise the naturalness of the speech in order to produce elongated vowel sections. In this analysis individual words are of the form 'consonant vowel consonant' (CVC) with the vowel held for approximately two seconds. Only the actual vowel sound was recorded from each word avoiding the areas of coarticulation.

The recording equipment used consists of a 486, 33 MHz Elonex Personal Computer with an Ultrasound Max soundcard employing a sampling rate of 22 KHz. An Audio Technica ATM73a head mounted microphone was placed to the side of the subject's mouth to reduce wind noise. Special acquisition software [1] which allows for windowing of the input data enabled the capture of clean vowel sections with no co-articulation regions. To reduce possible noise contamination the subject is placed away from the computer in a sound reducing booth. A complete description of the database is given in Appendix B.

[1] Phoneme acquisition software supplied by Alan Wrench of CSTR

In total 15 subjects were recorded, 10 male and 5 female. Each subject was asked to read a total of 60 words, in the CVC elongated format, which consisted of 12 different words each repeated 5 times. The exact placement of the words is given in Table 5.1.

1 heat (/i/)	13 caught (/O/)	25 heat (/i/)	37 hat (/{/)	49 hood (/U/)
2 hart (/A/)	14 head (/E/)	26 hurt (/U@/)	38 hate (/eI/)	50 hurt (/U@/)
3 hut (/V/)	15 hot (/Q/)	27 hit (/I/)	39 hoot (/u/)	51 hot (/Q/)
4 heat (/i/)	16 hat (/{/)	28 hood (/U/)	40 hat (/{/)	52 caught (/O/)
5 hart (/A/)	17 hate (/eI/)	29 hurt (/U@/)	41 hart (/A/)	53 head (/E/)
6 hit (/I/)	18 hoot (/u/)	30 hit (/I/)	42 hut (/V/)	54 hot (/Q/)
7 hood (/U/)	19 hat (/{/)	31 caught (/O/)	43 heat (/i/)	55 caught (/O/)
8 hurt (/U@/)	20 hate (/eI/)	32 head (/E/)	44 hart (/A/)	56 hate (/eI/)
9 hit (/I/)	21 hut (/V/)	33 hot (/Q/)	45 hut (/V/)	57 hoot (/u/)
10 hood (/U/)	22 heat (/i/)	34 caught (/O/)	46 hood (/U/)	58 hat (/{/)
11 head (/E/)	23 hart (/A/)	35 head (/E/)	47 hurt (/U@/)	59 hate (/eI/)
12 hot (/Q/)	24 hut (/V/)	36 hoot (/u/)	48 hit (/I/)	60 hoot (/u/)

Table 5.1: *The CVC words that the subjects were given*

The subjects, although they were mostly native English speakers, originated from a wide range of geographic areas in the UK and as such have a wide variety of accents. One subject is German although he has a good English speaking voice. Potentially this would be a problem since different accents produce different vowels from the same written word, a classic example being the Scottish pronunciation of 'hood' as /h/u/d/ (similar to 'food') rather than /h/U/d/ as in 'good'. Such ambiguities were noted prior to recording and different words chosen as required to ensure that the subject was producing the desired vowel. As a secondary check the position of each recording on a formant chart was examined, as shown in Figures 5.1 and 5.2 which show both a male and a female subject, to ensure that the desired vowel had indeed been produced. The formant charts were generated using XWaves formant analysis tools which are easily fooled into giving occasional errors. Examples of these can be seen on both charts, for instance the stray "heat" in Figure 5.1. Such points are easily checked manually but in some cases the test words themselves were ambiguous and therefore should be removed from the data set. These words were 'hate' which ranged from /e@/ to /i/,'hood' which ranged from /u/ to /V/ and 'hurt' which contained a rolling 'r' in many cases.

5.3 Emdedding the system

An elongated vowel, spoken at constant unforced volume, can be embedded into a d dimensional state space using either time delay embedding or SVD embedding

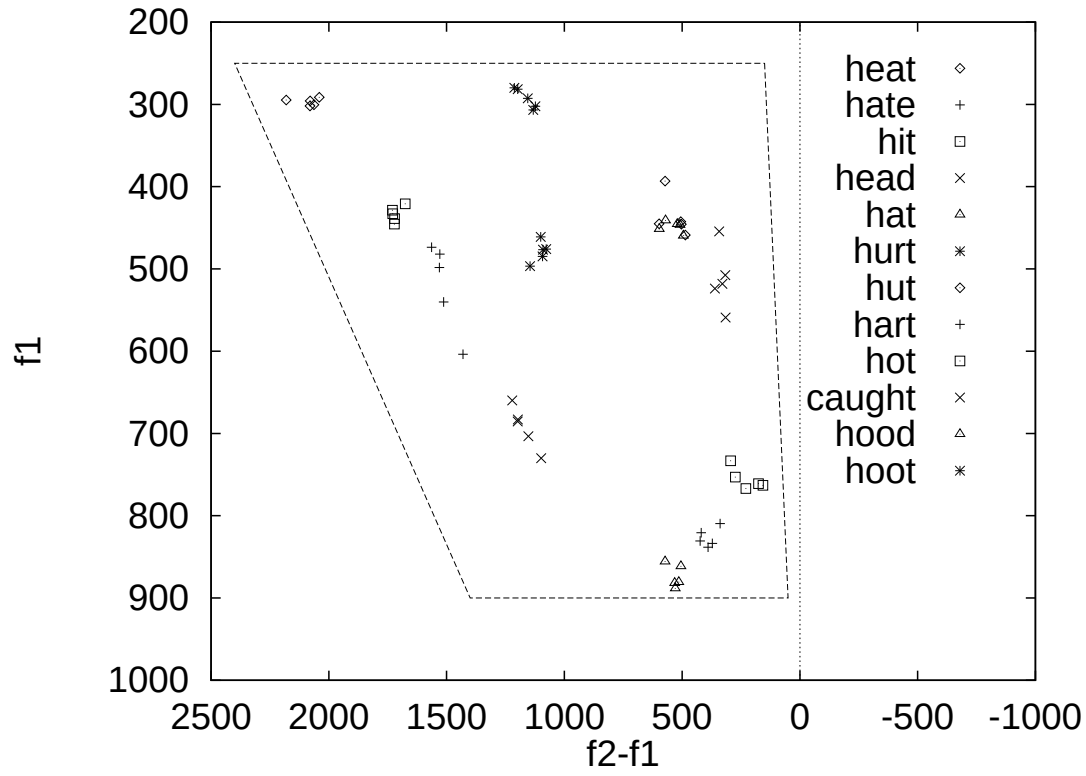


Figure 5.1: *Formant chart for speaker 'pb'.*

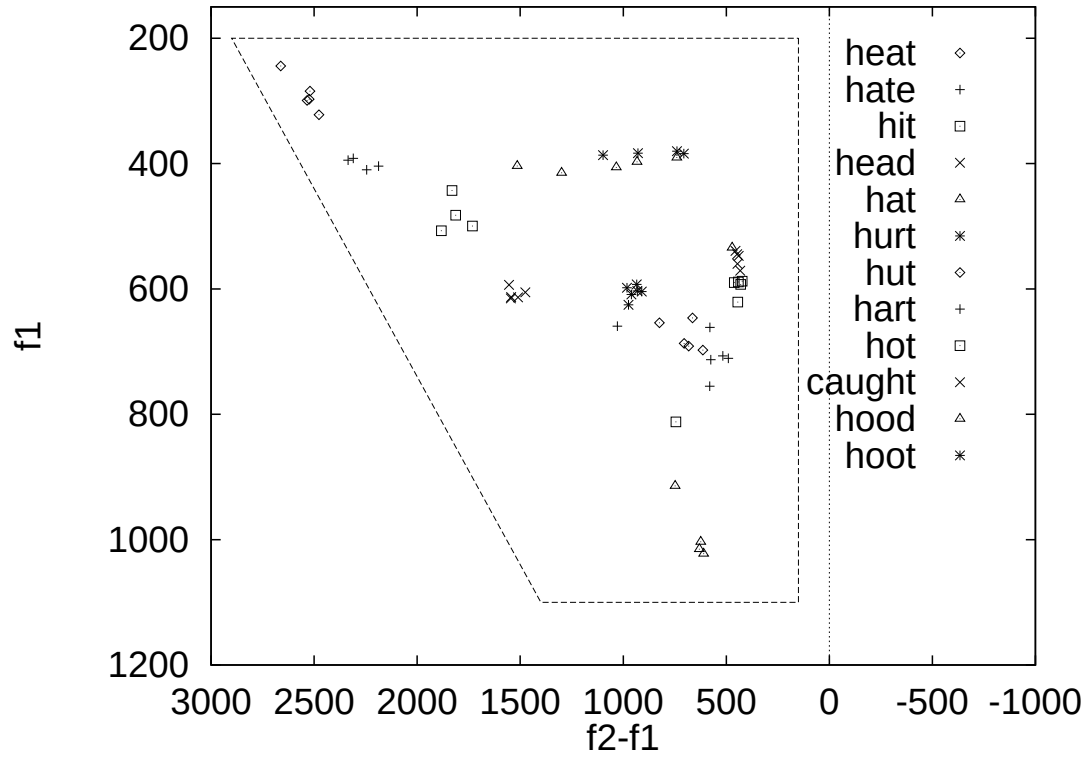


Figure 5.2: *Formant chart for speaker 'rw'.*

[118]. Mutual information gives an approximate optimal time delay for the embedding although any delay that unfolds the attractor is suitable [22,78]. Figures 5.3 and 5.4 show the mutual information for the vowel /i/ as in heat and the resulting attractors for a range of embedding delays. The mutual information suggests a delay of around 10 to 25 samples which can be seen from the attractors in Figure 5.4 to produce adequate unfolding of the trajectories leaving no self crossing points. It is worth stressing that on the two dimensional medium of paper the attractors may appear to have crossings but in the full three dimensional space this is not the case.

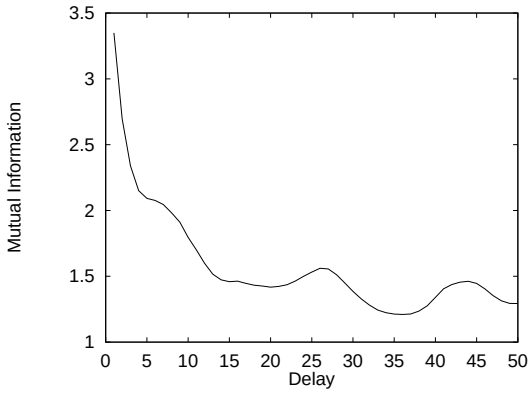


Figure 5.3: *Mutual information for the vowel /i/ for speaker 'pb'.*

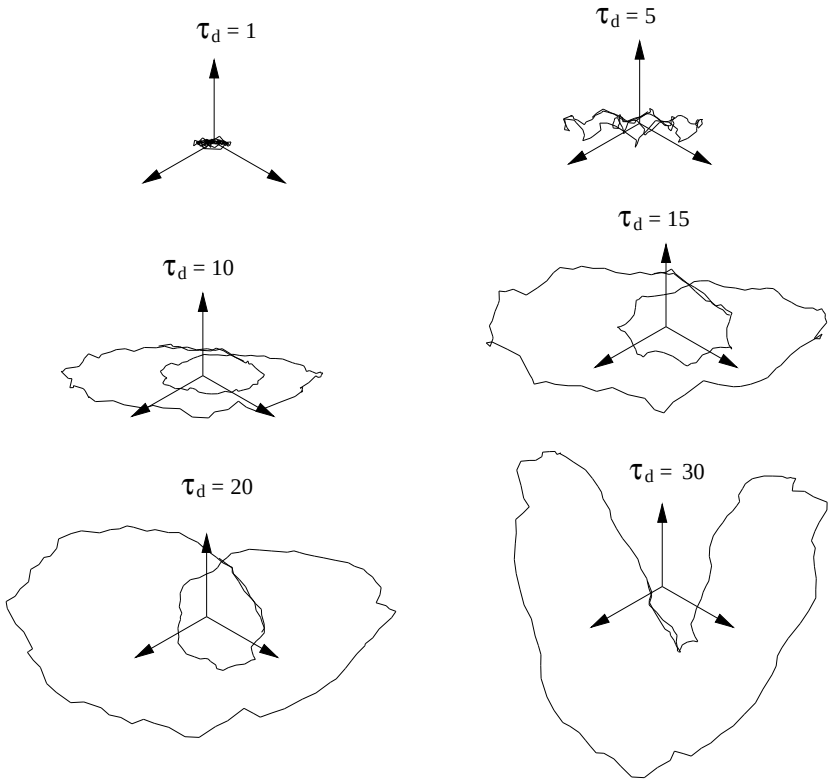


Figure 5.4: *Using the time delay to unfold the attractor.*

Figure 5.5 shows a three dimensional time delay embedding for the vowel /I/ as in *hit* where the state space vector $\underline{x}_i$ is formed from

$$\underline{x}_i = (x_i, x_{i+\tau_d}, \ldots, x_{i+(d-1)\tau_d})$$

with $\tau_d = 10$ samples, and the SVD embedded attractor using a window of 50 samples. The SVD embedding produces a far smoother attractor which is an indication that the noise components of the signal have indeed been reduced whilst clearly the underlying structure has been retained.

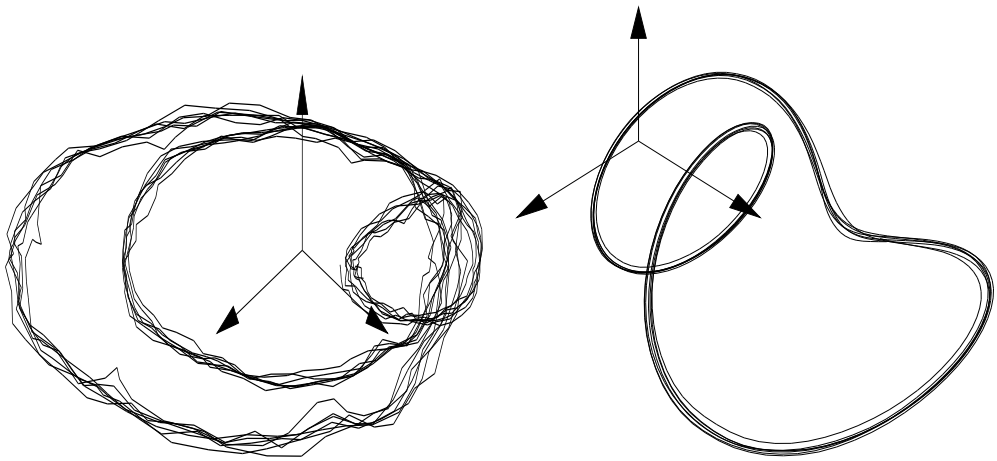


Figure 5.5: *Time delay and SVD embedding of [I] as in hit*

The resulting attractor for each vowel can then be plotted individually and placed into a formant chart. The resulting diagram, Figure 5.6, which is for time delay embedding, shows a number of interesting features. Firstly the attractors are all fully unfolded; that is they contain no areas where trajectories intersect in state space. This is important since it suggests that the attractors are low dimensional. Secondly there seems to be a correlation between the position in the chart (vertically) and the number of loops or folds in the attractor. Progressing anticlockwise around the chart the complexity of the attractor grows; more loops and folds become evident, until the position for 'hot' whereupon the complexity begins to decrease again. This is an interesting observation that does appear to correlate well with reports that the correlation dimension of a vowel depends, in a similar way, on its formant chart position [119].

Now that we have a low dimensional, fully unfolded attractor for each of the vowels we can analyse them using a number of different invariant geometrical measures which are covered in the next sections.

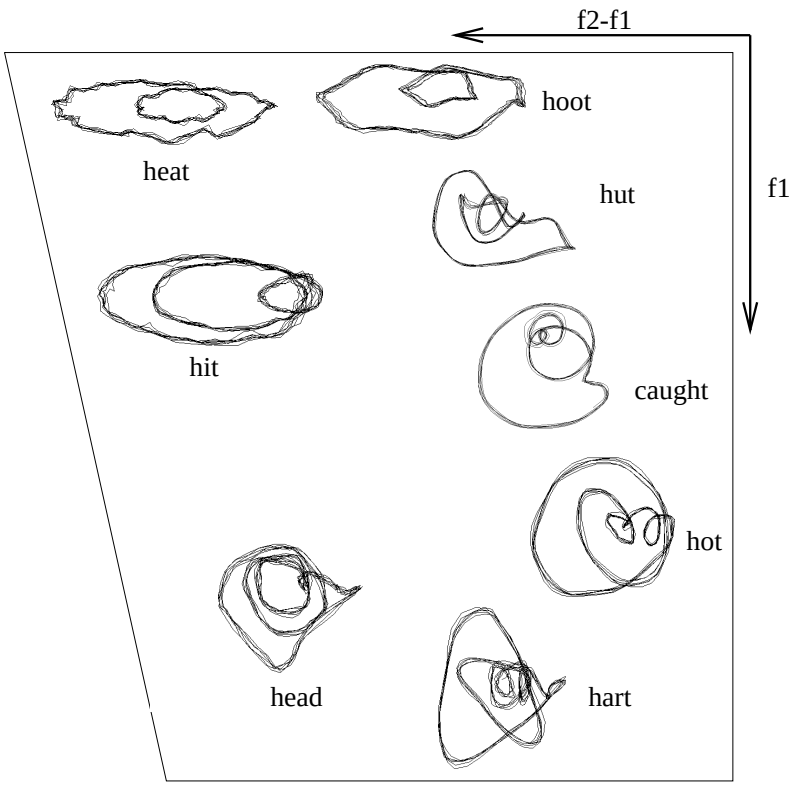


Figure 5.6: *Vowel attractors shown on a formant chart*

5.4 Short term prediction properties

Figure 5.7 shows the behaviour of the vowel /i/ as in 'heat' for a range of embedding dimensions. The curves show no significant improvement in the model for embedding dimensions greater than 7 or 8 suggesting that the actual system has a dimension that could, by Takens' theorem [62], be as low as 3 or 4. The 'flattening' of the curve for low k values is a sign that there is a level of background additive noise, or high dimensional behaviour, affecting the data. This level varies from vowel to vowel, as can be seen in the vowel summary shown in Figure 5.9, suggesting that this 'noise' is a property of the vowels themselves and not an external noise source that has corrupted the data; external background noise would show up in equal amounts on each vowel. One explanation for this would be that vowels have small amounts of fricative type noise added on top of the underlying low dimensional system; if you make the vocal tract configuration for most vowels and then stop voicing but continue to allow air to flow then a sound is still produced in the same way a normal fricative is produced, i.e. creation of turbulence. This 'noise' makes it difficult to assess the dimension from the gradient since it is unclear where exactly the gradient should be measured, as shown in Figure 5.8, however a very approximate value can be determined as shown in Figures

5.7 and 5.8. The gradient shown is approximately 0.71 which makes the information dimension $d_i = 3$, although it should be stressed that some areas of the curve, before the roll off, have lower gradients suggesting an information dimension of anything up to 4.

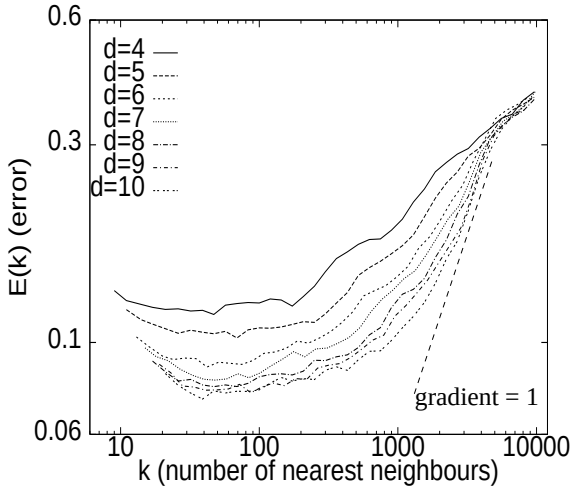


Figure 5.7: *Prediction errors for the vowel /i/*

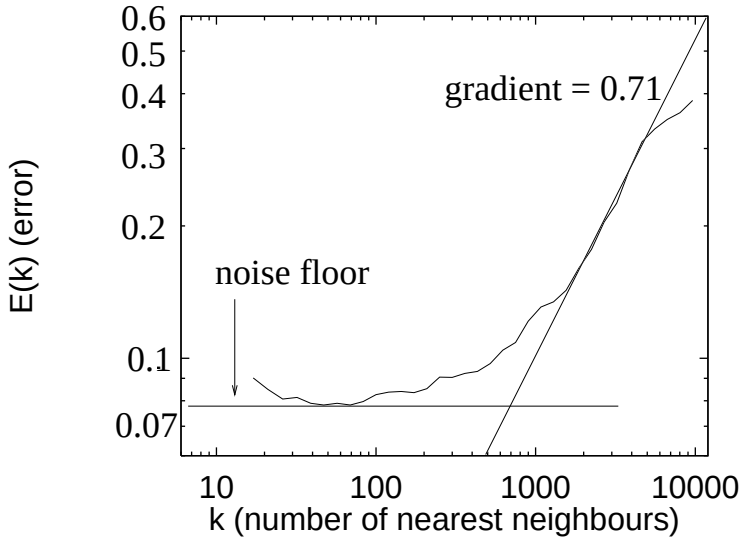


Figure 5.8: *Prediction errors for the vowel /i/ showing the gradient and fricative noise floor*

Figure 5.9 shows a summary of the curves for a range of vowels using an embedding dimension of $m = 7$. The vowels all seem to be of a similar dimension but with differing levels of apparent fricative noise according to the articulation of the sound.

As has already been stressed this technique is only very approximate in identifying a system's dimension and therefore other measures must be used to augment these

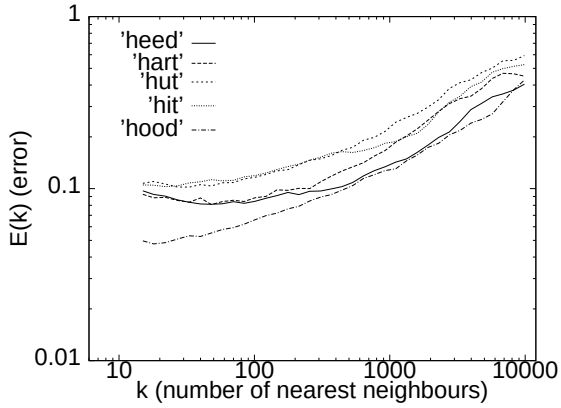


Figure 5.9: *Summary of prediction errors for a range of vowels*

results. An associated measure to this is the Lyapunov spectrum which directly gives a definition of the system convergence and divergence in state space and therefore the fundamental limits on the predictability of the signal. The calculation of Lyapunov spectra is a non-trivial problem which depends greatly on *a priori* information about the dimension of the system. The predictability measure used shows that the system has a *low* dimension but greater accuracy is required before attempting analysis of the Lyapunov spectra.

5.5 The underlying dimension of the system

To gain a full picture of the underlying system it is necessary to use as many different measures as possible. In this section we look very briefly at the correlation dimension and then at a technique known as local singular value decomposition analysis.

The correlation dimension is widely acknowledged to be difficult, if not impossible, to apply to real world data especially where noise is a factor [97]. In particular the calculation depends critically on the size of the data set being greater than 10^d. Since the maximum number of points in the data set is 40000 points with a possible dimension of as high as $d = 4$, the data falls right on the sufficiency border line. The levels of 'fricative' noise that are contained in the vowels further complicate the correlation dimension measurement. These problems can be clearly seen in Figure 5.10 which shows the correlation dimension calculation for the vowel /i/, as in heat: the main point of convergence is very small and indistinct and there is more than one point of convergence evident on the plot. That said it is still clear from the plot that the system is indeed low dimensional although exactly what dimension is not evident. Although

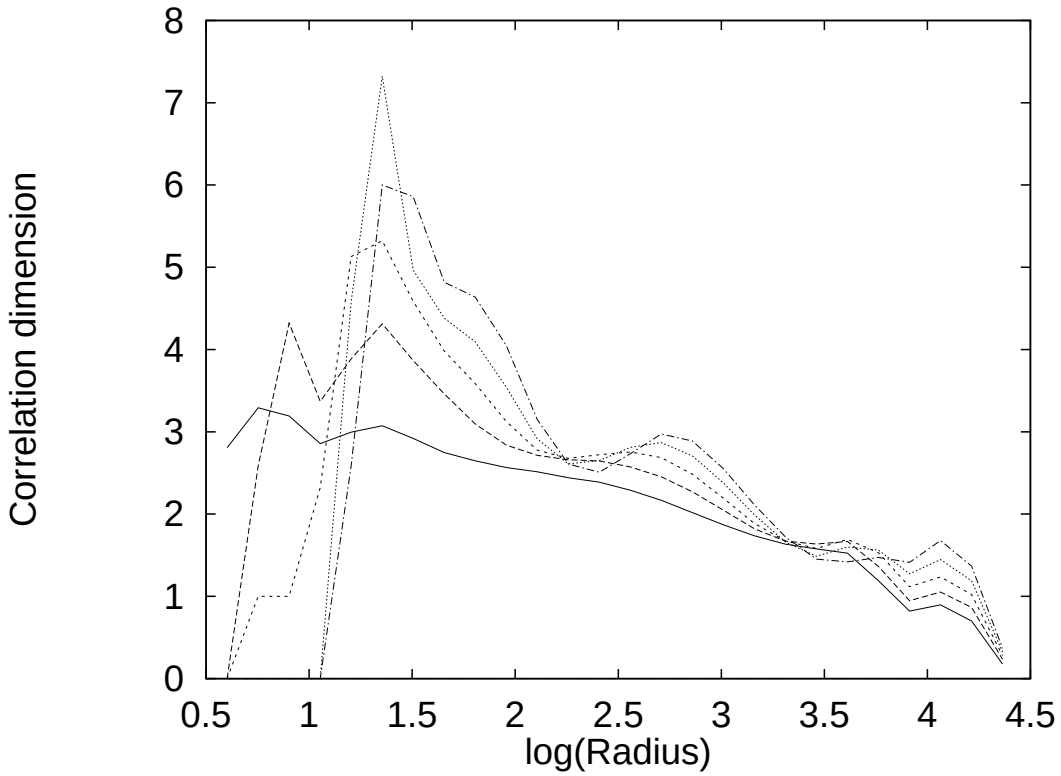


Figure 5.10: *Correlation dimension for the vowel /i/ using 35000 points, delay of 10 samples and embedding dimensions from 3 to 8*

the correlation dimension does gives some indication of the dimension of the system it is necessary to confirm the results by means of another measure; the local singular value decomposition.

Bearing in mind the previous results for simple systems, it is possible to see from Figure 5.11, which shows the results for the vowel /u/ as in 'hoot', what the dimension of a vowel is. It is possible that areas of the attractor may have a higher dimension than others and therefore it is important to build up a picture of the whole attractor and not just one small section, hence many test points should be used. The Figure shows just two examples at different points which, in this case, are representative of the attractor as a whole. In both cases, although it is clearer in the left panel, there appear to be two singular values which increase with the radius of the neighbourhood, suggesting that the system is no more than three dimensional. Analysis of the other vowels shows the same trend suggesting that the system has an underlying dimension of three.

Now that we have assessed the dimension of the system, and are confident that the dimension is of the order of three, it is possible to calculate the Lyapunov exponents.

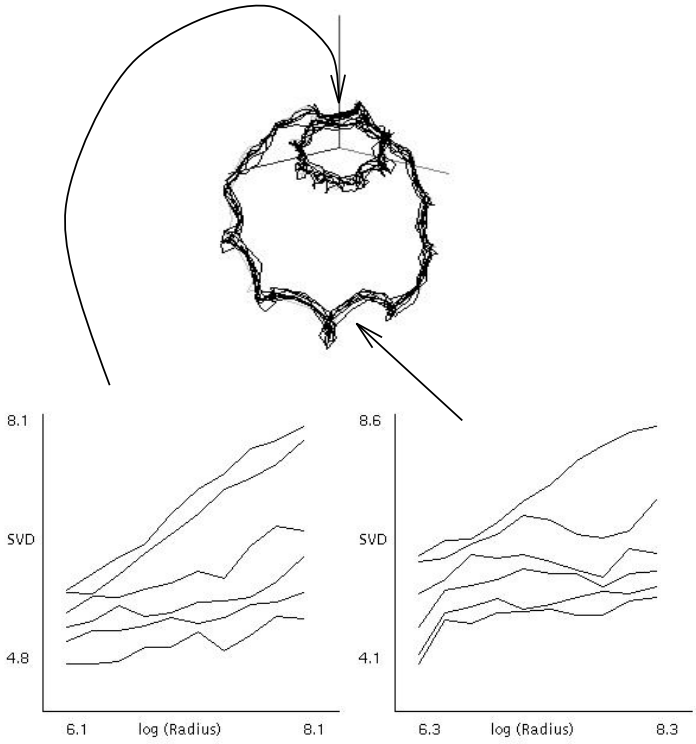


Figure 5.11: *Local Singular Value Decomposition analysis for the vowel /u/*

5.6 Lyapunov spectra analysis

A full description of the algorithm used to calculate the exponents, as given in Chapter 4, is not required at this stage but a re-iteration of the general outline is useful. The data is embedded into a global embedding dimension using singular value decomposition and reconstruction as already described. This embedding dimension is much greater than the expected dimension of the system so that the attractor is fully unfolded. The Lyapunov exponents are then estimated by using a trajectory matrix, often called a tangent map, as an approximation to the Jacobian, $J(p)$, as given in equation 3.15. The trajectory matrix is evolved around the attractor being renormalised using a QR-factorisation technique [67] once the divergence becomes too large. The average value of the eigenvalues of these trajectory matrices are then a direct measure of the Lyapunov exponents. In order to reduce the effects of noise on the calculation two techiques are applied: firstly the neighbourhood set is re-embedded into a lower dimension which is equal to the dimension of the system itself [29] to ensure that no spurious exponents are generated, and secondly the trajectory matrix is formed from a set of averages of displacement vectors rather than the vectors themselves. These have the effect of averaging out the noise whilst preserving enough of the dynamical

information to allow accurate calculation. The additional technique of data concatenation is not required since the subjects produced elongated vowels which contain sufficient samples to produce meaningful Lyapunov calculations.

The calculation requires a large number of parameters to be set at appropriate values for the data being analysed. The reason for this is quite simply that some of the parameters, such as the embedding window, level of noise reduction and the renormalisation period, depend on the properties of the data itself. For example noisy data will need more noise reduction at the expense of small scale accuracy. In order to get the best out of the algorithm the parameters have to be tried over a range of values and the optimal setting for each parameter chosen. This is a lengthy and time-consuming process which, in this case, with a large number of individual data, was accelerated by the use of a Cray [2] T3d supercomputer to get estimates of the best parameters to use.

The exponents were calculated over the full set of data for two of the subjects (randomly chosen, both male); pb and mc. In each case the parameters were varied over reasonable ranges suggested by the knowledge already given by the other measures. The selected results shown in Figures 5.12 to 5.14 are representative of the general behaviour of all the data giving the Lyapunov exponents in bits/second. The important points to note from these Figures are outlined below. At low SVD window lengths the noise is not fully removed as is evinced by the exponent which becomes zero for higher window lengths. Over the data set as a whole the longest the SVD window needs to be is approximately 50 samples, above this no benefit in noise reduction can be seen at considerable loss in computational speed. The size of the reinitialisation step seems to be best at mid-range values of around 10; too low and the noise is of the same scale as the expansion and therefore masks the true dynamics, too high and the expansion is limited by the size of the attractor. The embedding dimension plot shows that increasing the number of dimensions into which the system is locally embedded does not cause new positive exponents to appear or significantly alter the zero exponent. This would suggest that a local embedding dimension of 3 is sufficient for the system under examination. We have already seen that the system is low dimensional and the stability of the results at an embedding dimension of 3 further agrees with this premise.

The optimised parameters can now be applied to the full data set. The following is a summary of the results shown both as an average of the results for each individual

[2] thanks to EPCC for the use of the T3d Cray

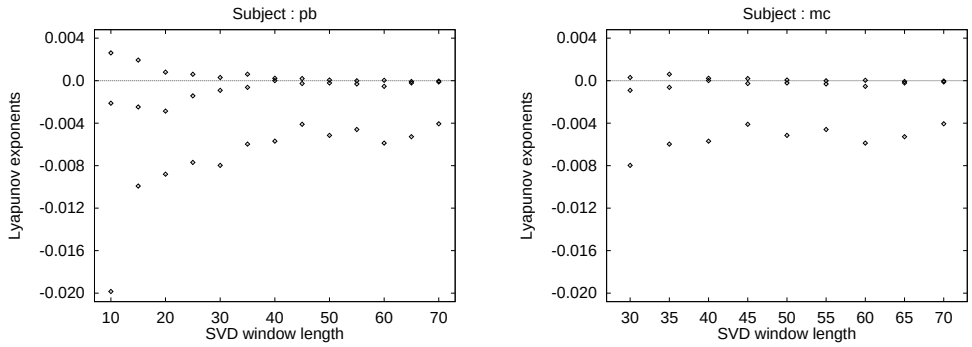


Figure 5.12: *Lyapunov exponents for the vowel /Q/ as in hot for a variable SVD window length. Other parameters are* 200 *neighbours;* 20 *vectors in the neighbouhood set;* 2000 *iterations of* 4 *evolve steps each.*

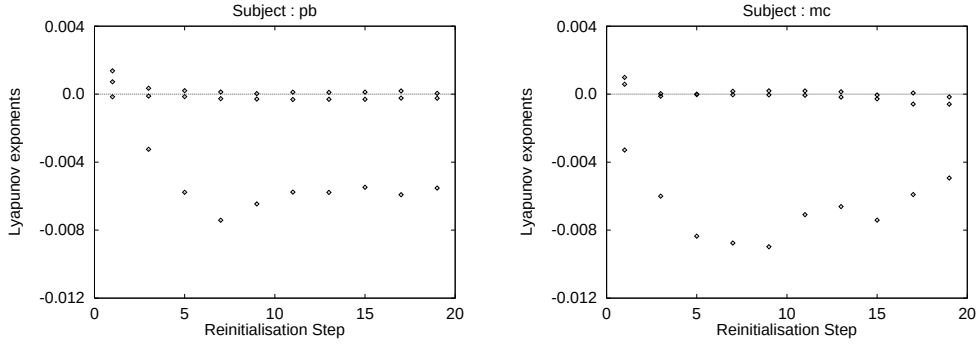


Figure 5.13: *Lyapunov exponents for the vowel /U/ as in hood for a variable reinitialisation step window length. Other parameters are* 200 *neighbours; 20 vectors in the neighbourhood set; SVD window of* 50*;* 2000 *iterations.*

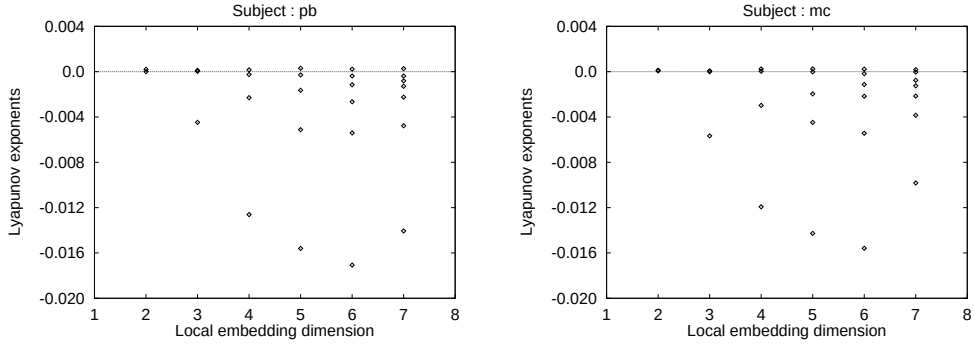


Figure 5.14: *Lyapunov exponents for the vowel /U/ as in hood for a variable embedding dimension. Other parameters are* 200 *neighbours; 20 vectors in the neighbourhood set; SVD window of* 50*;* 2000 *iterations of 4 evolve steps in each.*

subject, that is the average of 5 examples of a vowel for each person, and an average across all the subjects by repetition position of the vowel; that is the number of times that the vowel has already been said, e.g. from Table 5.1 number 10 is the 2nd repetition of hood (/U/). These averages ensure that the subjects are not changing the characteristics of the vowels over the recording period. The errorbars shown on the plots indicate the maximum and minimum values with the average marked by the diamond.

Several important features can be seen from the results shown in Figure 5.15 to 5.17. The average value in all cases is of the form of a zero and two negative exponents indicating that the data is non-chaotic. There is a small variation in the average value of the negative exponent from person to person. This can be shown not to relate to either the fundamental period or the volume of the vowel, as is clear from a comparison of male/female results; the female speech is quieter and higher in pitch than the male equivalent and yet there is no obvious difference in the exponents. One possible explanation of this variability is that different people do not use exactly the same articulation to produce the same vowel and it is this variability that is being seen. Indeed the size of the max/min spread that can be seen would suggest that even a single individual produces a variety of different levels of convergence for any vowel, consequently it is then natural to expect to see a variation from person to person also. By splitting the results into male/female it becomes clear that the male speakers seem to have a greater spread on the zero exponent. This cannot be explained by the variability in pronunciation since a spectrum without a zero implies a temporal variability over the attractor. However it can be explained by the actual recording process. The vowels are artificially elongated since the subjects had to hold the vowels for a much greater length of time than would normally occur in speech. Over this length of period small changes can, and do, occur in the articulation of the vowel and it is these that are being seen. It is interesting to note that males seem to have more variability than females which is quite consistent with observations made at the recording stage; females found it far easier to sustain constant pitch than did many of the males.

Overall the results do show conclusively that the vowels analysed are not chaotic, that is that they do not have a significant positive exponent, and that this conclusion is not affected by the local embedding dimension which has been examined from values of 3 to 7 with no significant additional exponents becoming visible.

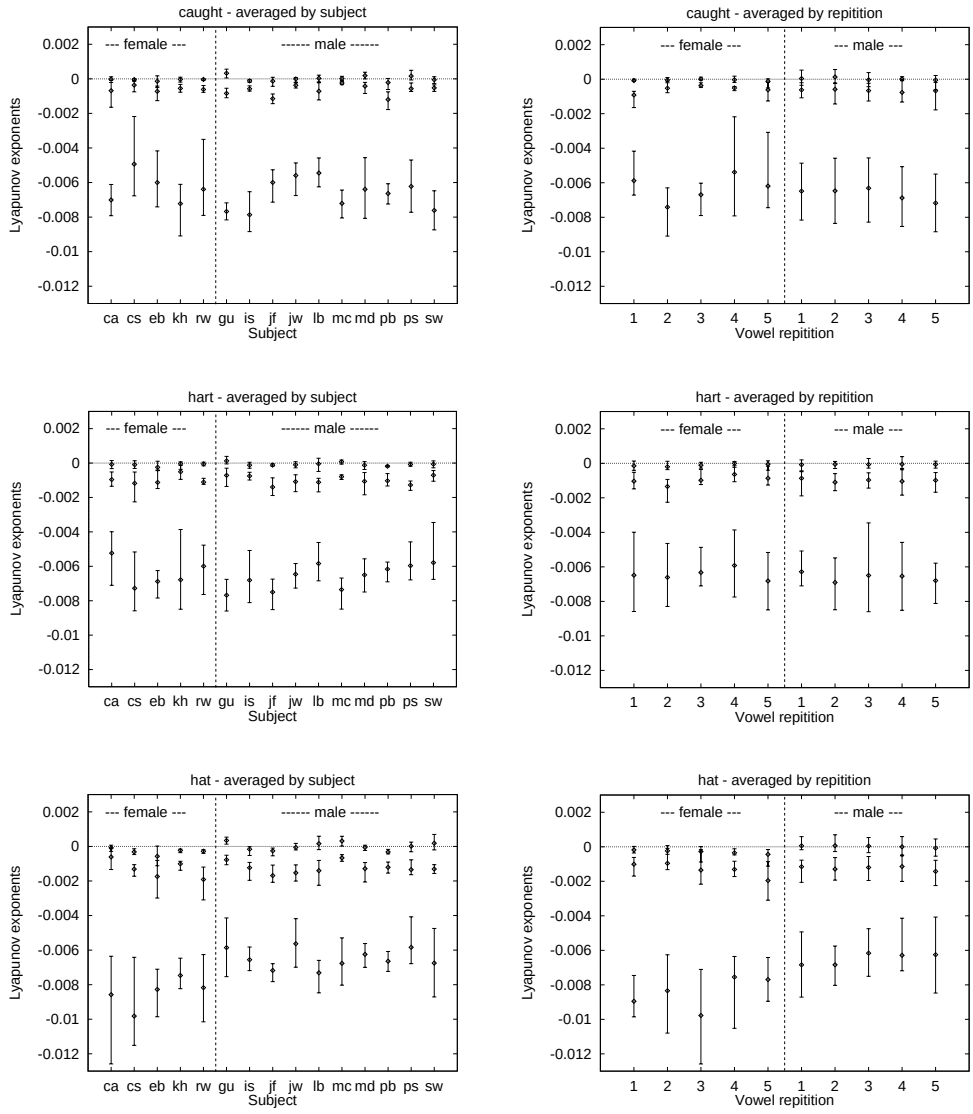


Figure 5.15: *Lyapunov exponents for the vowels /O/,/A/ and /{/ using the following parameters; SVD window length* 50*, global embedding dimension* 7*, local embedding dimension* 3,200 *neighbours forming 20 vectors in the neighbourhood set,* 2000 *iterations of* 10 *evolve steps each.*

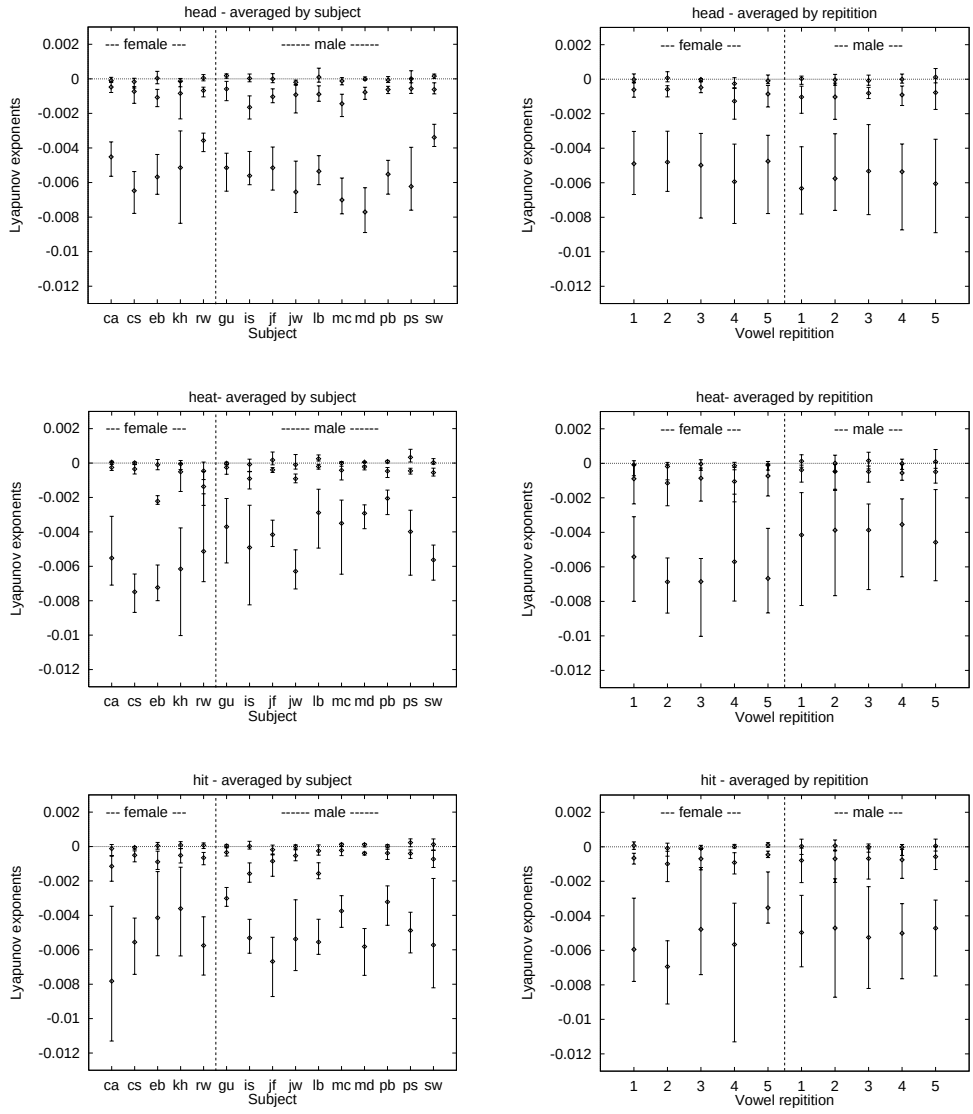


Figure 5.16: *Lyapunov exponents for the vowels /E/,/i/ and /I/ using the following parameters; SVD window length* 50*, global embedding dimension* 7*, local embedding dimension* 3,200 *neighbours forming 20 vectors in the neighbourhood set,* 2000 *iterations of* 10 *evolve steps each.*

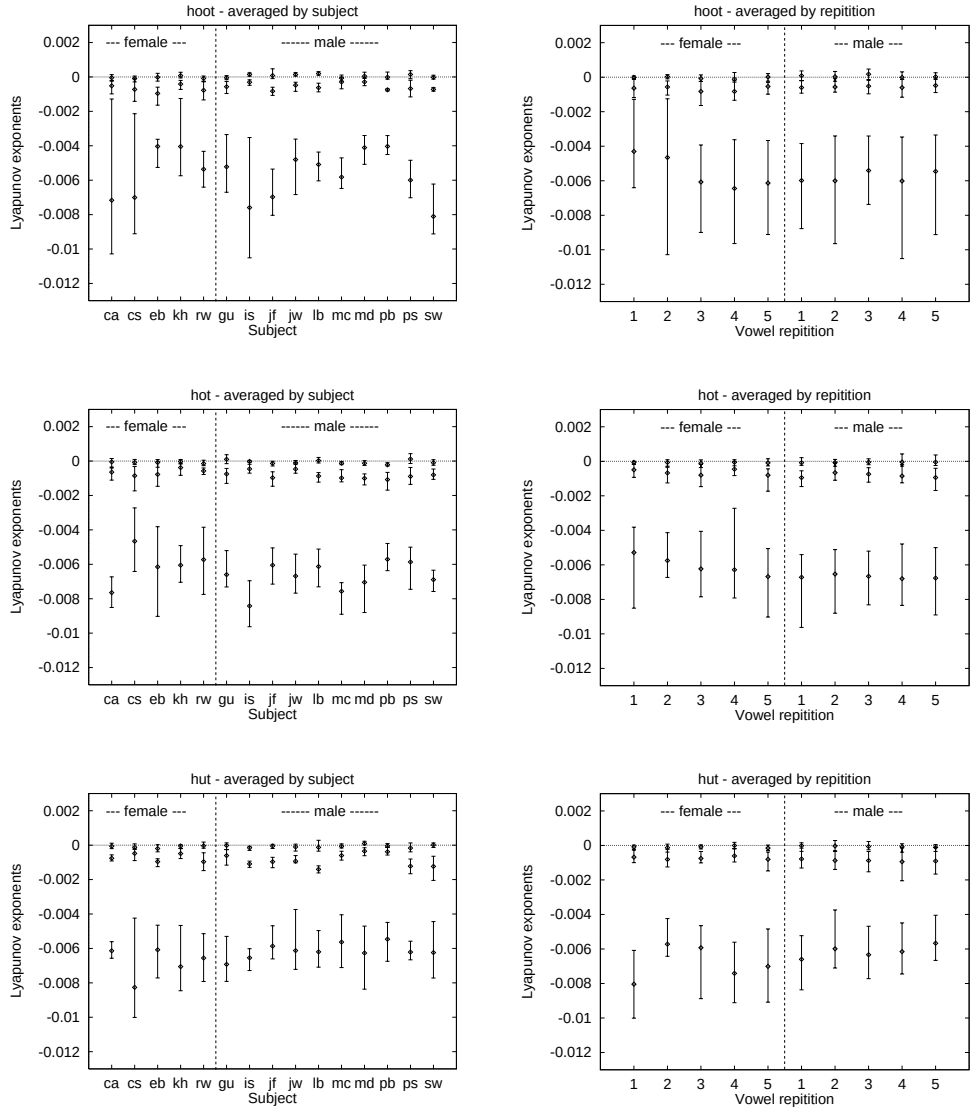


Figure 5.17: *Lyapunov exponents for the vowels /u/,/Q/ and /V/ using the following parameters; SVD window length* 50*, global embedding dimension* 7*, Local embedding dimension* 3,200 *neighbours forming 20 vectors in the neighbourhood set,* 2000 *iterations of* 10 *evolve steps each.*

5.7 Conclusions

This chapter has given a full detailed analysis of vowel sounds with respect to potential chaotic behaviour. The vowels were recorded from sustained examples taken from multiple subjects which were then analysed using a wide range of analysis tools from time delay embedding to Lyapunov exponents. The short term prediction properties of individual vowel sounds suggest nonlinear, low dimensional behaviour which through the use of Lyapunov spectra is shown to be non-chaotic.

Chapter 6

SYNTHESIS

Speech synthesis can be produced using many varied techniques from source filter approximations to cut and paste approaches. This chapter presents a novel technique[1], based on the nonlinear dynamics of speech, that can be used to improve the possible performance of a cut and paste concatenation synthesiser. It is demonstrated that the technique can be implemented effectively and used to produce high quality synthesis.

6.1 Introduction

Speech synthesisers today still lack the qualities that are needed to make them sound natural [120]. Some of the shortcomings are at the phonetic transcription and intonation stage but there are also problems with the actual underlying sounds that the synthesisers reproduce. This can most commonly be found by attempting to reproduce sustained vowels which often results in very mechanical sounds that lack emotion. An example of this is given in Figure 6.1 which shows a state of the art synthesised "eighteen"[2]. The final vowel has been elongated so as to stress the poor reproduction of the repeated vowel; the time domain plot shows the lack of variation over the period and the repetitive structure is clearly evident which gives rise to a 'buzzy' sound on listening.

Exactly what is missing from these sounds is not clear but one problem is the tendency to reproduce exactly the same sound each time it is required. If the dynamics of the signal, rather than the signal itself, could be used to model the speech then this problem would be avoided since the resulting output would change quite drastically depending on the starting conditions. This technique depends greatly on the dynamics of the signal and therefore it is important to investigate the nonlinear dynamical properties of speech before attempting any synthesis. This is an area that has received much interest recently with many authors reporting differing evidence for and against the

[1] Patent application GB 9600774.5
[2] Thanks to BT for supplying this example from the Laureate synthesiser

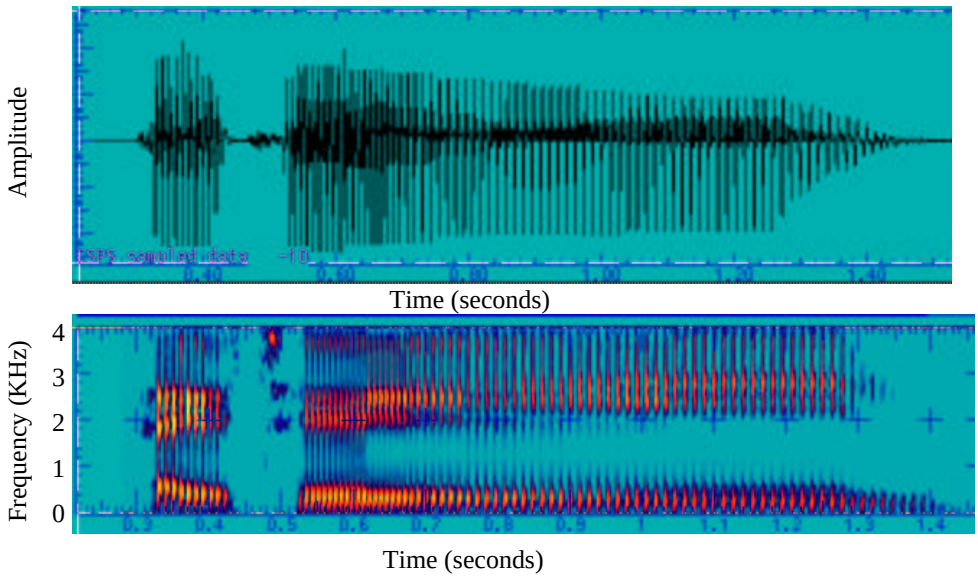


Figure 6.1: *Conventional synthesised "eighteen"*

existence of low dimensional attractors for speech [25, 26, 28, 83, 117]. In the previous chapter it was shown that there is evidence that speech is a low dimensional, nonlinear, non-chaotic system, and as such it should be feasible to use the dynamics as a synthesis tool.

6.2 The Algorithm

This section gives a full description of the underlying theory and a description of implementation details for production of a complete synthesis technique.

6.2.1 General Overview

Before describing all the details of how the synthesiser operates it is instructive to show a brief, and somewhat generalised, overview of a typical implementation. As shown in Figure 6.2 the synthesiser consists of a number of building blocks: a basic parser which converts input words into their constituent phonemes giving both duration and co-articualtion intervals; a general controller which decides which template to use and how to morph between different templates to create coarticulation and volume changes; and two routines which synthesise the next point according on whether the segment is voiced or not. In this particular example the synthesiser simply copies from

the template when the segment is not voiced.

6.2.2 Production of voiced segments

In the previous analysis the short term prediction and the Lyapunov spectra for isolated vowels were explored. The short term prediction properties show that a local model performs better than a global model and therefore the system can be considered as nonlinear. The results also show that the system is low dimensional, given by the gradient, and that the vowel sounds contain varying amounts of intrinsic fricative noise. A similar analysis [28] on fricatives shows them to be high dimensional, or stochastic signals. The Lyapunov spectra, for vowels, show that the system has no positive exponents and therefore is not sensitive to initial conditions (i.e. not chaotic).

The underlying dynamics of the system are obtained by using the state space representation of the time domain signal. Basically the system is embedded into state space using time delay embedding [85,121] and the nearest point to the initial start point for the synthesiser is located, as shown in Figure 6.3. It is then possible to estimate the dynamics of how points evolve onto the next step for a small localised area around that point, which naturally includes the start point itself. This estimate of the dynamics is then applied to the synthesised start point to produce a point one step ahead. The new synthesised point can then be viewed as a new start point and the process repeated thus building up as long or as short a segment as is required.

Since the dynamics are applied to the displacement vector, $\underline{b}_i = \underline{x}_j - \underline{x}_i$ as shown in Figure 6.3, rather than the actual vector, $\underline{x}_i$, it is important to ensure that $\underline{b}_i > 0$; if $\underline{b}_i = 0$ then it follows that $\underline{b}_{i+1} = 0$ and the synthesiser is merely performing a copying operation in state space. By thus ensuring that the chosen 'nearest point' is never actually coincident with an actual point on the stored template then the synthesised points will always produce a unique trajectory which is dependent on its exact starting position and the chosen 'nearest points'.

6.2.3 Morphing

When the synthesiser needs to move from one phoneme onto another then essentially a full set of intermediate attractors, or templates, between the two phonemes are required. For example, to move from /i/ to /A/ it is necessary to pass through /I/

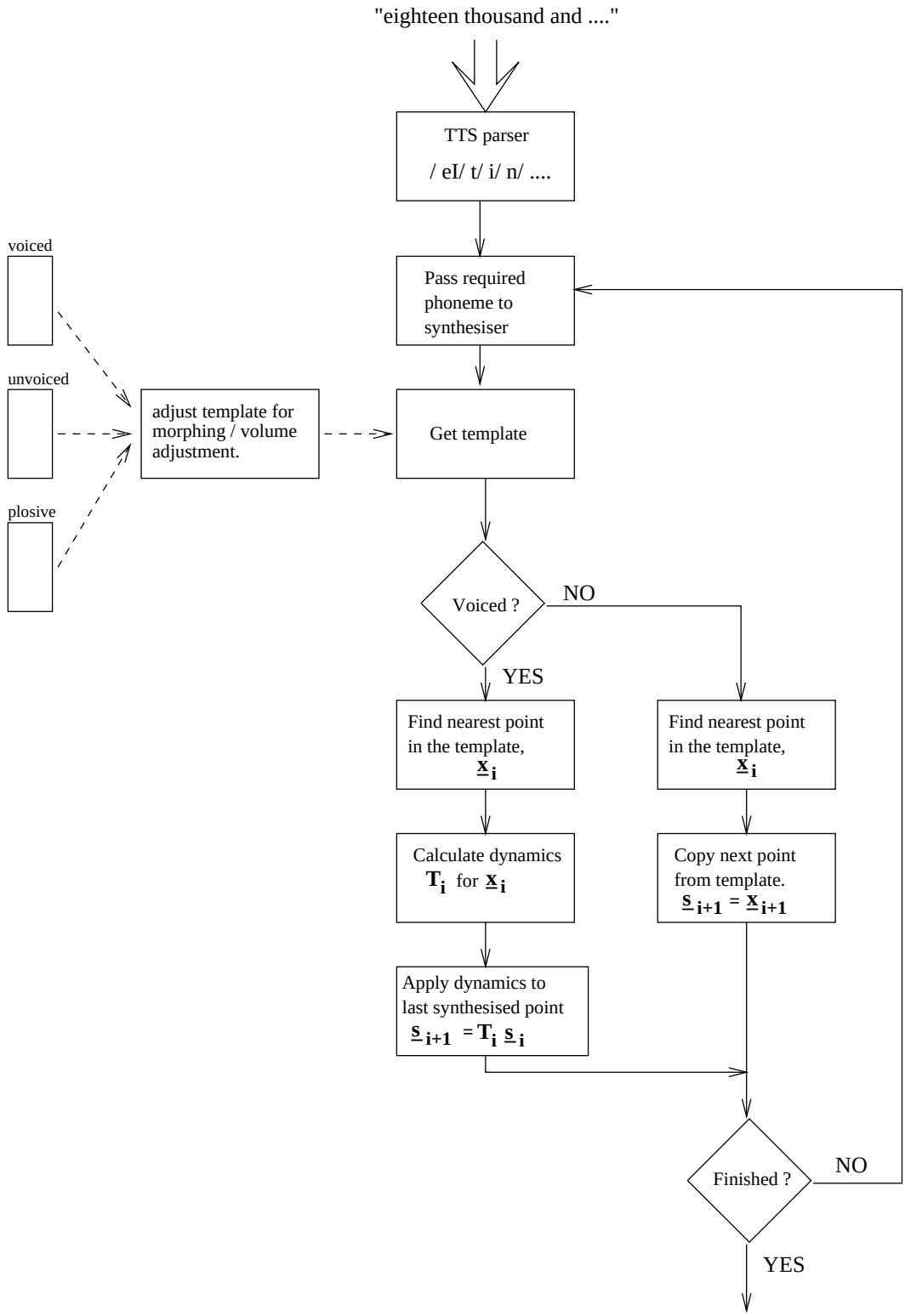
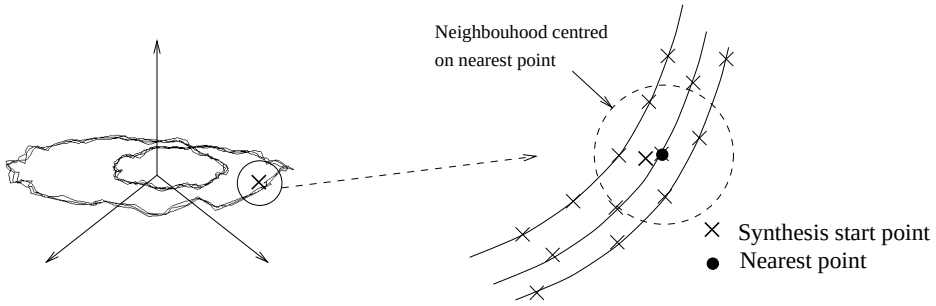
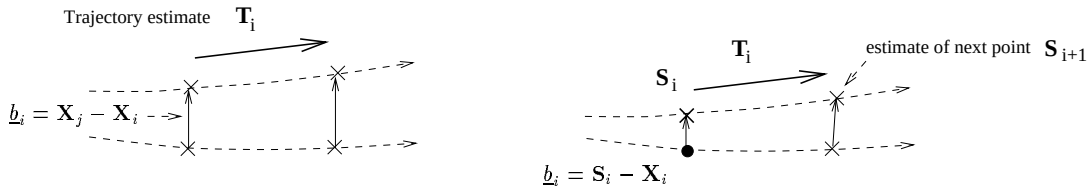


Figure 6.2: *Steps in the synthesis of a vowel*



FIND NEAREST POINT ------> FORM NEIGHBOURHOOD



$$\underline{b}_{i+1}^T = \mathbf{T}_i \underline{b}_i^T \qquad \mathbf{S}_{i+1} = \mathbf{X}_{i+1} + \mathbf{T}_i \underline{b}_i^T$$

ESTIMATE THE DYNAMICS ----> APPLY DYNAMICS TO LAST POINT

Figure 6.3: *Steps in the synthesis of a vowel*

and /E/ as seen on the formant chart, Figure 6.4. The result of using only the stored attractors is a discontinuity each time the template jumps from one phoneme to another which in practice is found to be too great, causing audible effects in the synthesised material. A natural extension would be to store a larger range of intermediate sounds but this is not really practical. Instead the intermediate attractors need to be constructed from the knowledge of the attractors that lie either side.

A simple and very effective approach is to produce an intermediate attractor $\underline{t}_i$ which is defined by

$$\underline{t}_i = (\underline{a}_i - \underline{e}_i)d + \underline{e}_i \tag{6.1}$$

such that $\underline{t}$ is the set of points that lie a particular fraction, d, of the Euclidean distance between corresponding points on the two main attractors, $\underline{a}$ and $\underline{e}$, as shown in Figure 6.5.

Once the intermediate attractor has been defined then the synthesis can be performed as before using the dynamics calculated from the new intermediate attractor. By using only small, and therefore often, incremental changes to d, the synthesised points never stray far from the template attractor that is being used and therefore no noticeable

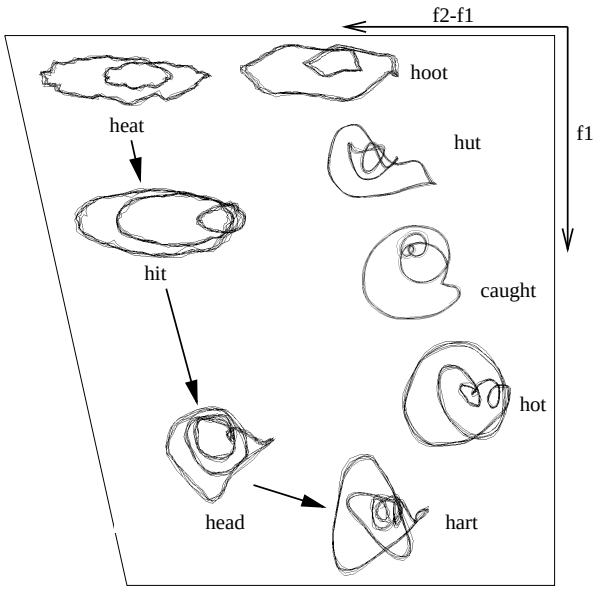


Figure 6.4: *Formant chart of phonemes*

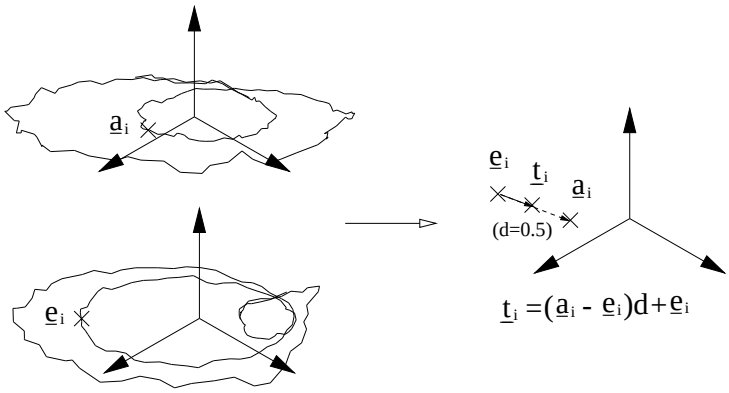


Figure 6.5: *Morphing between phonemes*

discontinuity occurs.

In order for this approach to work successfully it must be possible to locate 'corresponding' points on each attractor. If each cycle of the attractor were of constant length then corresponding points are simply those with equal time indexes. Unfortunately most people cannot hold perfect, steady pitch and therefore the recorded samples will contain a small amount of variation both in the average pitch and in the constancy of the pitch. It is therefore necessary to perform a pitch normalisation of the recorded samples before applying them to the synthesiser as described in the next section.

One further point that is worth noting is that the system should use all the information available. This means that where intermediate attractors exist in the database, as in the example of moving from /I/ to /a/, then these stored attractors are used as intermediate targets themselves. Thus to morph from /I/ to /a/ the system would morph from /I/ to /i/ and then /i/ to /e/ and then /e/ to /a/. This is found to be much more effective than trying the morph from /I/ to /a/ directly.

6.2.4 Normalisation

As already described it is important for the length of each cycle of the attractors to be exactly equal. This requires that some form of frequency normalisation be performed. Changing the fundamental frequency of a speech waveform is extremely difficult and is far beyond the scope of this section. Quite complex solutions such as PSOLA (Pitch Synchronous Over Lap and Add) [6,122] or Sinusiodal adding [123–125] can be implemented with realistic results but for the purposes of a simple synthesiser it is more efficient to use a simple stretching or squeezing of the individual cycles to be a particular length as shown in Figure 6.6.

This can be achieved by interpolating the waveform and then resampling each cycle, as defined by zero crossing points, such that each cycle contains the same number of samples, as shown in Figures 6.7 and 6.8.

6.2.5 Volume

In order for the synthesis to be realistic then some form of amplitude modulation needs to be considered so that volume changes can be affected. There are two main

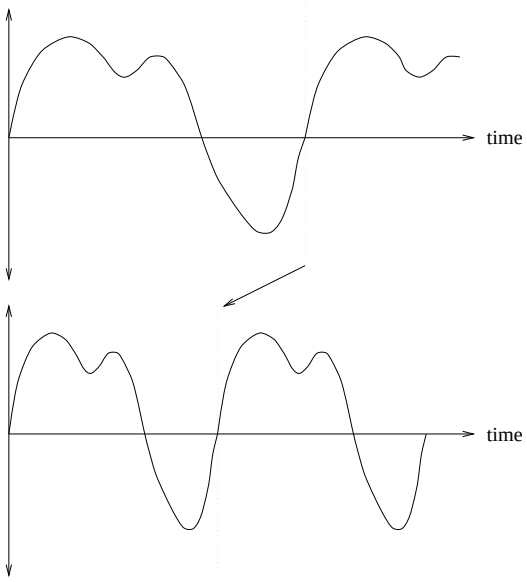


Figure 6.6: *Normalising the stored data*

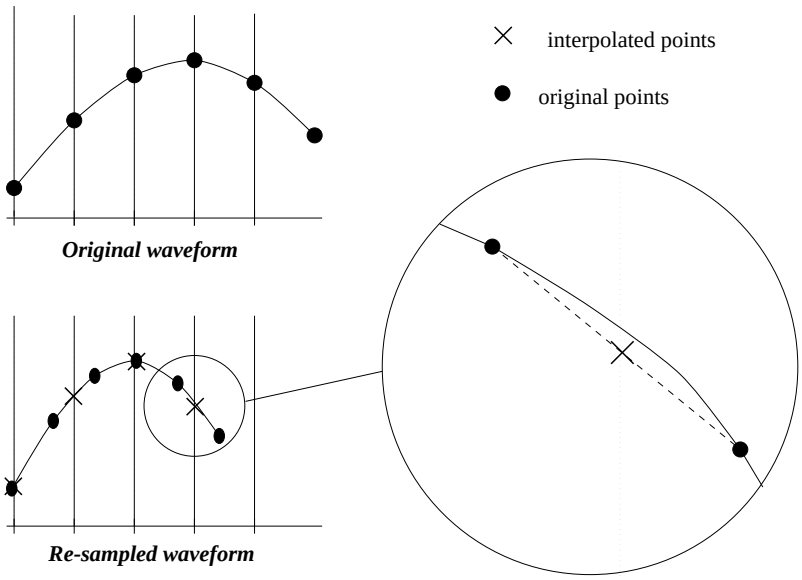


Figure 6.7: *Resampling the data*

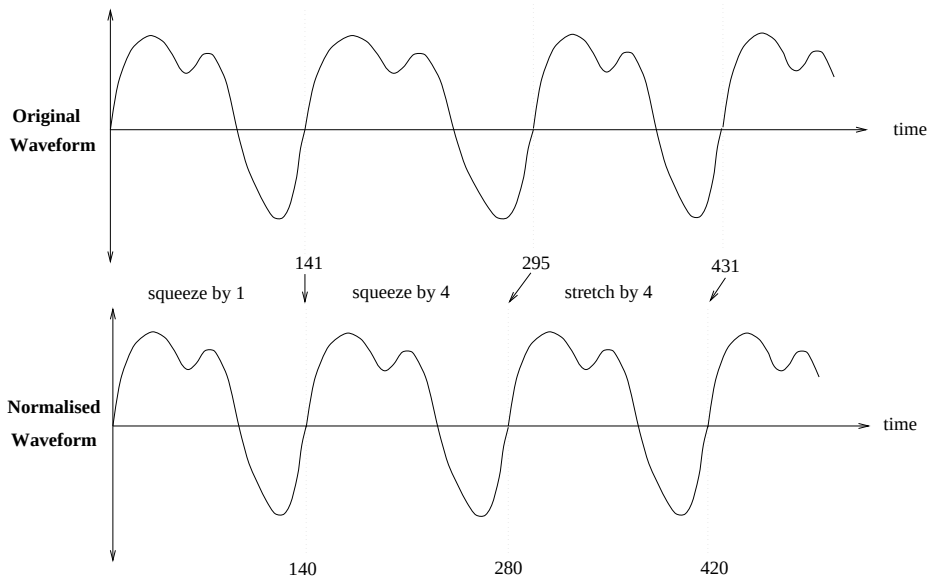


Figure 6.8: *Normalisation of the waveform*

approaches to this problem:

- 'Real time' modulation of the attractor templates,
- post-processing of the full-volume synthesised waveform.

'Real time' modulation of the attractors is in keeping with the general ethos of the synthesis technique since it works totally in state space. The basic idea is that if a morph can be produced to move between two different phonemes then by considering silence as a point attractor, at the zero origin, a similar morph can be used to move between silence and a phoneme as shown in Figure 6.9. Clearly this can be extended to allow for an increase in volume by morphing from the origin towards the phoneme. This can be extended even further by allowing $d > 1$ thus producing an effective increase in volume above the recorded level. This is potentially important for allowing stress and intonation to be included in the synthesis.

Although this technique is possible it does not allow for volume changes during transitions between phonemes. This can be overcome by having a general volume level which is set externally from the morphing procedure. This means that the stored templates are now defined by both the original template and a multiplication factor which can be varied to create all the intermediate attractors. This has the same effect as already described but now allows morphing between phonemes which have arbitrary volumes. Again it is important that increases in volume are made in many small steps rather than large jumps to ensure continuity.

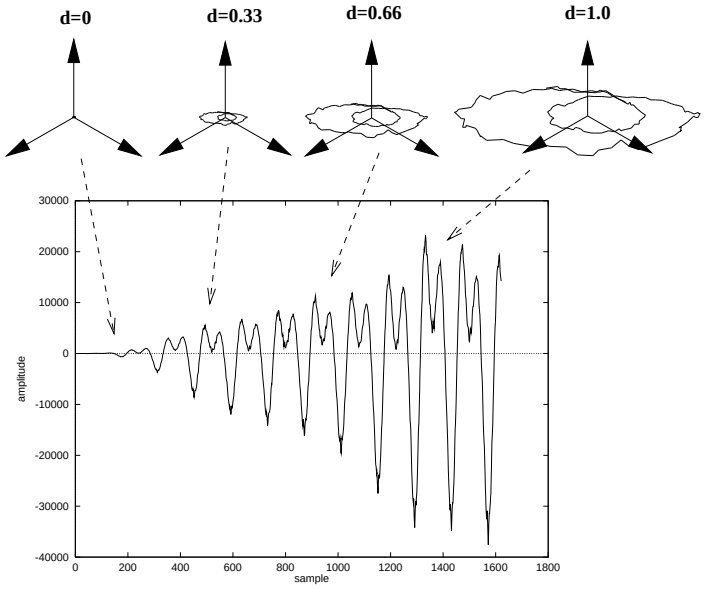


Figure 6.9: *Morphing from a silence to phoneme*

An alternative to this technique is to generate the synthesised waveform and then apply volume modulation as a post-processing operation as shown in figure 6.10.

On the whole there is little to choose between the techniques although it could be postulated that the post processing technique provides a structure for producing greater variability in the synthesised output. This variability would be created by allowing the start point to be chosen at random from the template attractor; the output will be multiplied by zero because the synthesiser starts from silence and therefore the choice of start point is not restricted. The random choosing of a start point would mean that each time the word is generated a significantly different realisation would occur.

6.2.6 Pitch variation

Pitch variation is more of a problem than volume variation because of the strict constraints placed on the fundamental frequency by the morphing techniques, namely the need to normalise all the attractors. This rules out the possibility of performing the same morphing technique that is used for volume modulation since corresponding points in the attractors could not be identified. To recap such a morph generates all the required intermediate attractors from a combination of the original and the desired attractors. Fortunately a full set of the intermediate attractors of different frequencies can be defined from any one of the stored templates by resampling the time domain waveform to the required number of samples per cycle, in the same way as for norm-

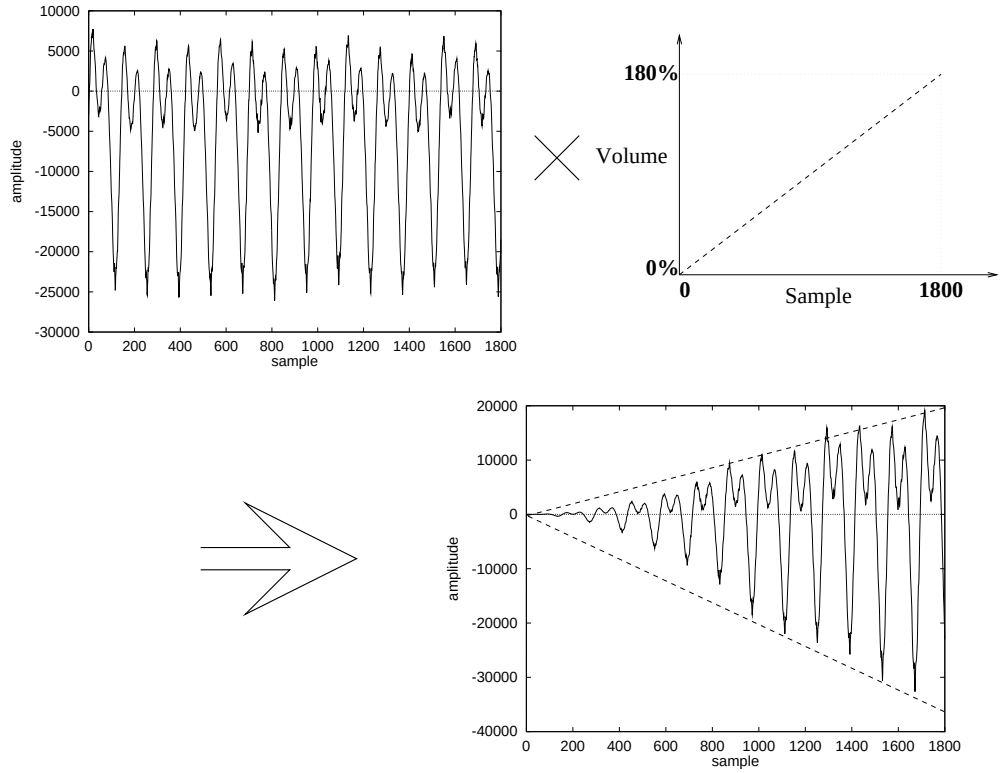


Figure 6.10: *Post-processing approach to volume modulation*

alisation. This is shown graphically in figure 6.11 which shows a number of attractors for the vowel /I/ which have been resampled to four new fundamental frequencies. So long as the step size between the templates is small enough then the synthesised points will never be too far from a point in the next template and therefore should not cause discontinuities.

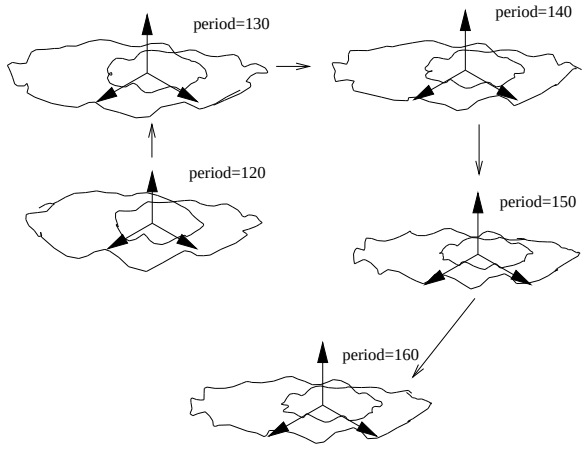


Figure 6.11: *Changing the fundamental frequency*

It should be noted that the technique of resampling a waveform to produce a different fundamental frequency is by no means ideal and is certainly not realistic. A full

study of more realistic techniques is not within the scope of this thesis although it should be pointed out that *any* technique based in the time domain could be applied to the generation of intermediate time waveforms from which the attractors are then generated.

6.3 System Appraisal

The synthesiser presented in this chapter is meant as a demonstration that the underlying waveform synthesis technique works and does offer possible advantages over conventional techniques. It is important to bear this in mind when appraising the system since the synthesiser is a very basic one, using no complicated phonetic description or intonation and a very simple resampling technique to produce fundamental frequency shifts. These shortcomings are naturally reflected in the resulting speech and so it is important to focus on the underlying aim which is to produce elongated segments of speech which sound natural, that is that they do not possess the 'buzziness' common in other techniques, and that simple coarticulation between phonemes is possible.

The test words chosen to examine the effectiveness of the technique are a simple set of numbers. These are used because they contain a range of important properties : elongated vowels as in eighteen where the /i/ vowel is sustained, examples of vowel to plosive and plosive to vowel transitions, examples of fricative to vowel transition and examples of vowel to nasal and nasal to vowel transitions.

Table 6.1 shows the SAMPA transcriptions of the test set along with a list of suggested durations for each phoneme[3].

The data given in Table 6.1 is for a straight concatenation synthesiser and so gives no data on either the volume or period of co-articulation between phonemes. Furthermore it also highlights the need for diphthongs, such as /eI/ in "eight", where two vowels slide continuously into one another. Such a procedure is not possible in a straight concatenation approach but is feasible using the synthesis technique described earlier. Thus the duration values, and indeed the phonemes themselves are used as a rough guide only.

[3] Thanks to Mike Edgington of BT labs for supplying this data

word	SAMPA	duration(ms)
one	/w/	72
	/V/	76
	/n/	145
two	/t/	177
	/u/	217
three	/T/	94
	/r/	86
	/i/	173
four	/f/	61
	/O/	252
five	/f/	93
	/aI/	135
	/v/	167
eighteen	/eI/	90
	/t/	102
	/i/	107
	/n/	145

Table 6.1: *The test words used for system appraisal*

The first word synthesised is "eight" which is created as shown in figure 6.12, with the resulting waveform and spectrogram analysis shown in figures 6.13 and 6.14 which also shows an example of a real "eight" for comparison. Clearly the real example

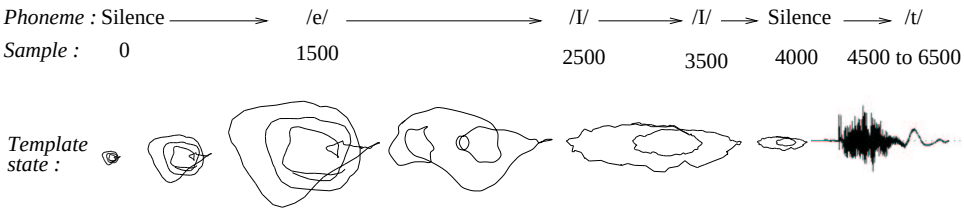


Figure 6.12: *Steps in the generation of the word "eight"*

is rather longer in duration than the synthesised example but it still serves as a very useful benchmark for the synthesised version. Several important features can be seen from the spectrogram plots :

- the basic formant structure has been accurately reproduced.
- during the morphing period the formants of the two phonemes appear to overlap rather than gradually moving from one set to the other, as seen in the real "eight".
- there are no excessive discontinuities such as would be audible as clicks.

Of course the proof is in the eating and therefore to really find out how good the synthesised version is it must be listened to. Obviously this is not possible within

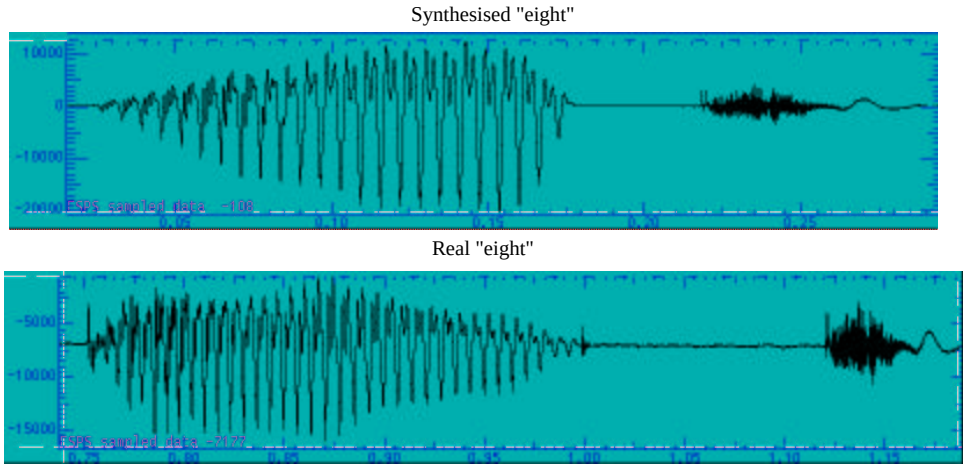


Figure 6.13: *Time domain plots of a synthesised "eight" and a real "eight"*

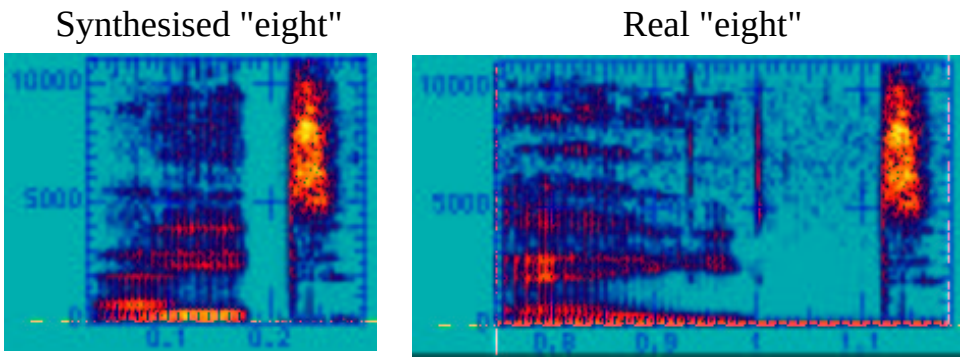


Figure 6.14: *Wideband spectrograms of the synthesised "eight" and a real "eight"*

the confines of the text of this thesis but a selection of the synthesised words in .WAV format have been included on disk supplied and documented in Appendix C. Listening to the synthesised eight two things become apparent; firstly the word is too short and secondly the reproduction of the diphone, /eI/, is extremely smooth and realistic. This would suggest that the overlapping of the formants, in this case, works quite well but it should be stressed that further work needs to done to show whether this is generally true.

The second word tested is the word "one". This proved an extremely difficult word to reproduce since the move from silence to /u/ and then from /u/ to /Q/ seemed to produce an unexpected /m/ sound at the start of the word creating "mone" rather than "one". The solution to this was to include an extra target template in between /u/ and /Q/ which makes sense since the formant trapezium traversal requires both /U/ and /O/ be passed through to get to /Q/. The full description of the steps used is shown in figure 6.15 and the resulting waveforms and spectrograms in figures 6.16 and 6.17.

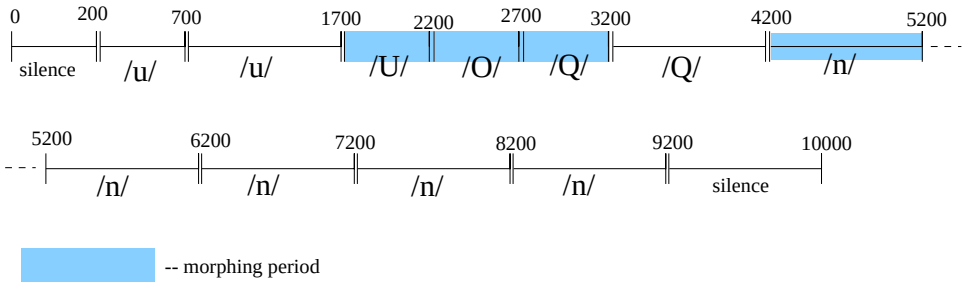


Figure 6.15: *Steps in the generation of the word "one"*

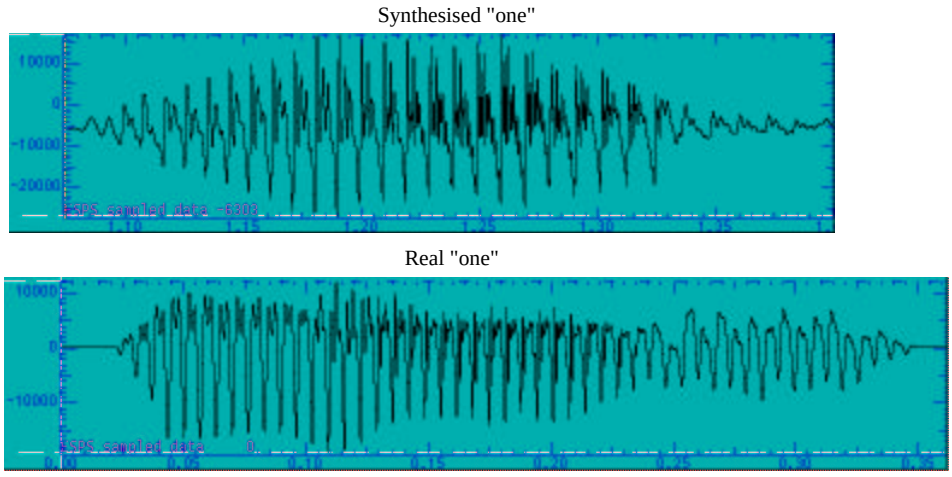


Figure 6.16: *Time domain plots of a synthesised "one" and a real "one"*

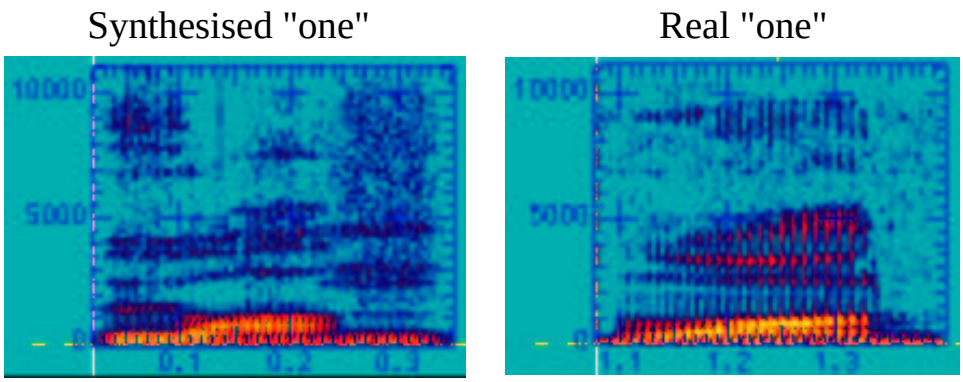


Figure 6.17: *Wideband spectrograms of the synthesised "one" and a real "one"*

This time the two waveforms look substantially different especially in terms of the amplitude envelope. This was found to be necessary since simply emulating the real world resulted in a poor synthesised word. By allowing a much quicker volume increase at the start the spurious /m/ problem was reduced and it was found that the nasal /n/ needed to be quite loud to be heard significantly. Even though the waveforms do look quite different they do in fact yield similar spectrograms: the formant structure is clear in both. The main difference between the two is at the end where the synthesised nasal seems to contain a small amount of high frequency formant structure not present in the real one. The cause of this is not known and further work is required on the subject but it is possible that the reason is that the nasals are of a different dimension or contain chaotic properties which would be revealed on a full analysis similar to that performed on the vowels.

Other examples of synthesised numbers are given on the disk with one in particular which is worth mentioning. The word "three" requires a small amount of rolling 'r' in order to sound like "three" and not "thwee". The closest phoneme to this is /U@/ as in hurt where the 'r' is allowed to roll slightly. Unfortunately, as is evident from the synthesised "three", the database was not constructed with this in mind and so the sound is closer to "her" than "hurt" resulting in a word that sounds more like "thwee".

Since one of the aims of the synthesiser is to reproduce high quality elongated vowels then a natural progression is to add an elongated "een" on the "eight" to produce "eighteen". This entails the synthesiser being able to start from a pre-determined point in state space, as defined by the end of /t/, and then produce a non-repeating /i/. Figure 6.18 shows both the spectrogram and the waveform for a portion of the synthesised "eighteen". It is clear that the vowel is not simply repeating since variations are obvious in both the time domain and the spectrogram and indeed upon

listening the word sounds natural and contains none of the characteristic buzz evident in other forms of synthesis.

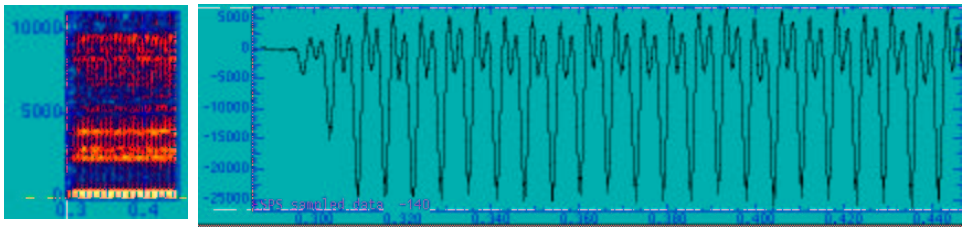


Figure 6.18: *Spectrogram and time domain plots of a synthesised "eee"*

6.4 Conclusion

In this chapter a novel synthesis technique has been proposed which makes use of the local, low dimensional, nonlinear dynamics of vowels to produce a synthesiser capable of high quality, natural speech including elongated vowels. The underlying theory of the synthesiser is explained and a working demonstration is detailed along with a number of example synthesised words. From the demonstration it is clear that the technique has much potential although a large amount of further work is required to fully integrate the technique into a fully operational system.

Chapter 7

CONCLUSION

This thesis has presented an analysis of the possible chaotic behaviour of vowel sounds using novel analysis techniques and has shown how these results may be extended to produce an innovative synthesis technique. This chapter discusses in detail the main achievements of the work including the significant development of robust analysis tools, analysis of a comprehensive database of vowels and a description, and demonstration of a novel synthesis technique based on ideas taken from nonlinear dynamics. The chapter then examines the possible areas of further work that could build on this analysis.

7.1 Achievements of the work

The achievements of this work are threefold : firstly the development of robust chaotic analysis tools; secondly the successful application of these tools to a database of vowels; and thirdly the development of a novel synthesis technique which draws from the results of the analysis and ideas taken from nonlinear dynamics and chaos theory.

Analysis of chaotic systems has only recently begun to move out of the laboratory and into the real world with the realisation that noise and data length are of paramount importance [111]. One of the greatest problems faced by the chaotic analyst is that the analysis tools almost always give results that are open to interpretation and in many cases lead to confusion or misleading conclusions. This has been particularly apparent with the application of chaos analysis to speech where authors have claimed dimension measurements[1] that range from as low 1.2 up to as high as 4 or 5 [18, 20, 22–28, 53, 58], and Lyapunov spectra that seem to both prove and disprove chaotic behaviour [22,23,25,28]. The problem is particularly acute because of the nonstationary nature of continuous human speech; the articulators in the mouth are continually adjusting to make the next phoneme and are seldom constant for more than 10ms. For

[1] a range of different measures are used

any analysis of speech to be truly complete it really needs to take into account both this nonstationary nature of continuous speech and the distinct levels of background noise that exist. Chapter 4 presented a range of tools that can be used to achieve such an analysis. It was shown in detail that current Lyapunov spectra algorithms are not robust and two important modifications were postulated that improve the performance dramatically. Firstly by forming the neighbourhood matrix using an average of a number of neighbourhood vectors, rather than a large matrix using the same vectors, an unparalleled noise robustness is achieved. Secondly a technique was described which allows an attractor to be described through the composition of a number of small sections of multiple attractors, all of which arise from the same set of system conditions, thus overcoming the problem of data length. These modifications were demonstrated using a chaotic Lorenz system with variable levels of additive noise and data length showing that they out perform the conventional Darbyshire and Broomhead technique [29]. It is important to stress that without these modifications such an analysis of speech would simply not be possible, with any confidence, so the impact of these modifications cannot be stressed enough.

In Chapter 5 the analysis tools described above were applied to speech. As described in a number of areas in this thesis, speech encompasses a wide range of different types of sounds, most of which are not suitable for this style of analysis; fricatives have been shown to exhibit quite high dimensional behaviour and are extremely difficult to produce long stationary segments, plosives are nonstationary by very definition, voiced fricatives contain levels of fricative noise which swamp the potential underlying voicing and again are difficult to produce in long stationary segments. The most suitable set of phonemes for chaotic analysis are vowels, which have been tentatively shown by others to have low dimensional [18,20,22–28,53,58], nonlinear characteristics. Consequently it is vowels that this thesis focuses on.

The data set used was a database of 15 people, 5 female and 10 male, each of which produced 5 repetitions of 12 different vowels that were chosen to spread over the whole vowel trapezium. Using the modified analysis tools described, these vowels were shown to have the following characteristics :

- optimal time delay of between 10 and 20 samples for time delay embedding. This is shown through use of the mutual information and examination of 3 dimensional phase space plots.
- evidence of nonlinearity. The short term prediction properties of the vowels

show a gradient of approximately 0.7, which translates to a nonlinear system with an information dimension of around 3, and a clear noise floor which is attributed to levels of fricative noise on the vowels. This level of frication is shown to vary over a range of different vowels.

- an underlying system dimension of 2 or 3. It is shown that the use of correlation dimension is not sufficiently conclusive and that an analysis of the local singular value spectra shows behaviour consistent with that of a 3 dimensional, or lower, system.
- non-chaotic Lyapunov spectra. The vowels were analysed using the noise robust technique described earlier and exhibit the following properties: a general structure of a zero and two negative exponents as consistent with a three dimensional, non-chaotic system; a small, but significant, spread in the value of the most negative exponent which is shown to be independent of both pitch and volume and is therefore attributed to the natural spread in articulation of similar vowels; males exhibit greater spread on the zero exponent than females do, which is attributed to the recording technique where most males found it very difficult to maintain stationarity over the whole 2 second speech segment.

Overall the analysis of the data set shows conclusively that vowels show low dimensional, nonlinear, non-chaotic characteristics.

Given that the vowels are low dimensional and not sensitive to initial conditions, as shown by them not being chaotic, then it is not unreasonable to attempt to utilise the dynamics of speech to produce a novel synthesis technique. This was shown in Chapter 6 which presents a novel synthesis technique which is based entirely in the state space rather than the conventional synthesis techniques which reside in either the time or the frequency domains. The technique recreates the dynamics of voiced speech from a number of templates of phonemes which are embedded into a three dimensional state space through time delay embedding. The dynamics are recreated by finding a neighbourhood of nearest neighbours which are iterated to allow the dynamics of the local space to be calculated. These dynamics are then applied to the last synthesised point to produce the next synthesised point. By repeating this process it is possible to create as long a segment of a vowel as is required and it is shown that the synthesised vowel has realistic level of variation leading to a more natural sound. It was shown, through a number different techniques such as template morphing, normalisation, pitch and volume control, that a complete synthesiser is feasible using

this technique. Results from a simple version of such a synthesiser were presented and show that the synthesised speech is both natural sounding and capable of realistic coarticulation effects.

7.2 Further work

The analysis presented in this thesis has been confined to vowels but a similar analysis might be feasible for unvoiced sounds if the problems highlighted earlier can be overcome. The main problem is that the fricatives appear to be quite high dimensional, which with the current analysis tools is not going to lead to conclusive results particularly because of the problems of data length which become more important the higher the dimension becomes. However, as has already been shown, chaos theory is still in its infancy and it is quite possible that, with the combination of increased research and increased computational power, a meaningful analysis may be possible in the future. This would certainly be of considerable use since it would enable the synthesis technique described in Chapter 6 to be extended to include all types of sounds not just voiced ones. This leads quite conveniently into the second area of further work which is the synthesiser itself.

The synthesiser described in this thesis is intended as a demonstration that the underlying theory can be applied successfully and is in no way meant to be a final packaged version. Most of the features that have been described to extend the synthesiser so that it can cope with pitch and volume changes, and even the morphing between templates, are all very simplistic solutions which could be looked at more thoroughly. It should be noted though that such an undertaking is an enormous one which would take many years to complete; most synthesisers currently available have had whole teams working for many man-years to get to their current states. Perhaps the most interesting area of research would be to look at exactly how the template attractors change with fundamental frequency. The current technique described uses a resampling approach in the time domain which has been acknowledged to be a non-optimal solution. A better solution may be to find a similar approach to the template morphing described which enables the intermediate attractors to be calculated. Again, a word of caution is worthwhile here since there are a number of potential problems; as the fundamental frequency changes so the optimal time delay embedding changes, creating the need for a variable time delay; using extra templates for each phoneme (several spread over

a range of frequencies) substantially increases the data storage requirements of the system. The last comment about system requirements is a topic that has been avoided thus far; is the system feasible both in terms of storage and real time processing speed?

It is difficult to assess the actual performance limitations of the synthesiser purely from the demonstration model because it was never designed to be quick or efficient, however it does serve as a starting point. The demonstration system is capable of resynthesising at approximately 10 samples a second, which is nowhere near real time; the speech is sampled at 22.050kHz which means that the synthesis speed needs to be around 22000 samples a second! Fortunately there are a number of potential speed-ups available:

- fast nearest neighbour routines. Currently the system uses very simple search routine to find the nearest neighbours. A similar routine in the Lyapunov exponent algorithm showed an increase in speed of over 50 times by being coded using an n-dimensional doubly linked list structure.
- precalculating the dynamics of the templates. Currently the dynamics are calculated each time a nearest point is located. This involves a considerable computational overhead which could be reduced if the dynamics for each point on the template were precalculated and stored. This obviously does have a down side which is that the storage requirements are vastly increased and also it would require a completely new morphing routine which morphed the dynamics rather the templates.
- removal of the graphical user interface. The demonstrator has no text parser to generate the rules controlling the synthesis which are currently defined by the operator through a graphical user interface (GUI) which would be redundant in a full system. Also the plotting of the trajectories into a three dimensional state space is a considerable overhead which is not required in a normal system.

With just the implementation of the ideas given above it is not unreasonable to assume that rates of 5000 samples a second are obtainable. This is bringing the system well into range of feasibility, especially bearing in mind the rate of increase in computer speed.

7.3 Summary

To finish off the thesis it is worthwhile reiterating the motivation and achievements of this work. The motivation comes from previous work which has shown that speech may contain low dimensional behaviour that could be exploited to improve synthesis techniques. The achievements of the work are that it has: produced significant modifications to the conventional tools of chaotic analysis, allowing a meaningful analysis of noisy data signals to be undertaken; given a comprehensive analysis of the potential chaotic properties of vowel sounds showing them to be low dimensional, non-linear and most importantly, non-chaotic; demonstrated a novel synthesis technique, based solely in the state space domain, which is capable of producing high quality, natural speech.

References

[1] A. Breen, "Speech synthesis models: a review," *Electronics and Communication Engineering Journal*, pp. 19–31, February 1992.

[2] G. Fant, *Acoustic theory of speech production*. Mounton and Co, 1960.

[3] S. K. Palmer and J. House, "The development of dynamic voice source rules for synthesis," *Proc of the Institute of Acoustics*, vol. 14, no. 6, pp. 577–584, 1992.

[4] D. H. Klatt, "Review of text-to-speech conversion for english," *Journal of the Acoustical Society of America*, vol. 82, pp. 737–793, September 1987.

[5] M. Edgington, A. Lowry, P. Jackson, A. P. Breen, and S. Minnis, "Overview of current text-to-speech techniques: Part i - text and linguistic analysis," *BT Technology Journal*, vol. 14, pp. 68–83, January 1996.

[6] M. Edgington, A. Lowry, P. Jackson, A. P. Breen, and S. Minnis, "Overview of current text-to-speech techniques: Part ii - prosody and speech generation," *BT Technology Journal*, vol. 14, pp. 84–99, January 1996.

[7] T. Haji, S. Horiguchi, T. Baer, and W. J. Gould, "Frequency and amplitude perturbation analysis of electroglottograph during sustained phonation," *J Acoust Soc Am*, vol. 80, pp. 58–62, July 1986.

[8] R. J. Baken and F. Orlikoff, "The effect of articulation on fundamental frequency in singers and speakers," *Journal of voice*, vol. 1, no. 1, pp. 68–76, 1987.

[9] J. Schoentgen and R. deGuchteneere, "An algorithm for the measurement of jitter," *Speech Communicatiop*, vol. 10, pp. 533–538, 1991.

[10] Y. Horii, "Fundamental frequency perturbation observed in sustained phonation," *Journal of Speech and Hearing Research*, vol. 22, pp. 5–19, March 1979.

[11] T. Kobayashi and H. Sekine, "Statistical properties of fluctuation of pitch intervals and its modelling for natural synthetic speech," in *ICASSP '90*, pp. 321–324, IEEE, 1990.

[12] P. Maragos, T. F. Quatieri, and J. F. Kaiser, "Speech nonlinearities, modulations, and energy operators," in *ICASSP '91*, pp. 421–424, IEEE, 1991.

[13] L. Hodgson, M. E. Jernigan, and B. L. Wils, "Nonlinear multiplicative cepstral analysis for pitch extraction in speech," in *ICASSP '90*, pp. 257–260, 1990.

[14] T. V. Ananthapadmanabha and G. Fant, "Calculation of the true glottal flow and its components," *Speech Communication*, vol. 1, pp. 167–184, December 1982.

[15] T. Koizumi, S. Taniguchi, and S. Hiromitsu, "Two-mass models of the vocal cords for natural sounding voice synthesis," *Journal of the Acoustical Society of America*, vol. 8, pp. 1179–1192, October 1987.

[16] H. M. Teager and S. M. Teager, "Evidence for nonlinear sound production mechanisms in the vocal tract," in *Proc NATO ASI on Speech Production and Speech Modelling*, pp. 241–261, 1990.

[17] G. Kubin, "Nonlinear processing of speech," in *Speech Coding and Synthesis* (W. B. Kleijn and K. K. Paliwal, eds.), pp. 557–610, Amsterdam: Elsevier, 1995.

[18] N. Tishby, "A dynamical systems approach to speech processing," in *ICASSP '90*, pp. 365–368, IEEE, 1990.

[19] L. Wu, M. Niranjan, and F. Fallside, "Fully vector-quantised neural network-based code-excited nonlinear predictive speech coding," *IEEE Trans on Speech and Audio Processing*, vol. 2, pp. 482–489, October 1994.

[20] P. A. Moakes and S. W. Beet, "Recurrent radial basis functions for speech period detection," *Proceedings of the Institute of Acoustics*, vol. 16, no. 5, pp. 271–278, 1994.

[21] S. S. Narayanan and A. A. Alwan, "Strange attractors and chaotic dynamics in the production of voiced and voiceless fricitives," in *EUROSPEECH '93*, pp. 77–80, 1993.

[22] H. P. Bernhard and G. Kubin, "Detection of chaotic behaviour in speech signals using fraser's mutual information algorithm," in *13th GRETSI symposium on signal and image processing*, 1991.

[23] H. P. Bernhard and G. Kubin, "Speech production and chaos," in *13th GRETSI symposium on signal and image processing*, 1991.

[24] P. Maragos, "Fractal aspects of speech signals: dimension and interpolation," in *ICASSP 91*, pp. 417–420, IEEE, 1991.

[25] S. McLaughlin and A. Lowry, "Nonlinear dynamical systems concepts in speech analysis," in *EUROSPEECH '93*, pp. 377–380, 1993.

[26] H. F. V. Boshoff and M. Grotepass, "The fractal dimension of fricative speech sounds," in *COSMIG '91*, pp. 12–16, IEEE, 1991.

[27] P. S. McDowell and S. Datta, "The fractal characterisation of isolated human speech," *Proceedings of the Institute of Acoustics*, vol. 16, no. 5, pp. 247–253, 1994.

[28] S. S. Narayanan and A. A. Alwan, "A nonlinear dynamical system analysis of fricative consonants," *The Journal of the Acoustical Society of America*, vol. 97, pp. 2511–2524, April 1995.

[29] A. G. Darbyshire and D. S. Broomhead, "The calculation of Lyapunov exponents from time series data." Submitted to Physica D, 1995.

[30] R. Paget, *Human Speech*. Kegan Paul, Trench, Trubner and Co., 1930.

[31] R. Linggard, *Electronic synthesis of speech*. Cambridge University Press, 1985.

[32] J. N. Holmes, *Speech synthesis and Recognition*. Van Nostrand Reinhold(UK), 1988.

[33] E. L. Reiegelsberger and A. K. Krishnamurthy, "Glottal source estimation: methods of applying the lf-model to inverse filtering," in *ICASSP '93*, pp. II542 – II545, IEEE, 1993.

[34] D. Y. Wong, J. D. Markel, and A. H. Gray, "Least squares glottal inverse filtering from the acoustic speech waveform," *IEEE Trans on Acoustics, Speech and Signal Processing*, vol. 27, pp. 350–355, August 1979.

[35] B. Cranen and L. Boves, "Aerodynamic aspects of voicing: Glottal pulse skewing revisited," in *ICASSP '85*, pp. 1085–1089, IEEE, 1985.

[36] I. R. Titze, "The physics of small-amplitude oscillation of the vocal folds," *J Acoust Soc Am*, vol. 83, pp. 1536–1552, April 1988.

[37] D. M. Brookes and P. A. Naylor, "Speech production modelling with variable glottal reflection coefficient," in *ICASSP 88*, pp. 671–674, IEEE, 1988.

[38] A. P. Lobo and W. A. Ainsworth, "Evaluation of a glottal arma model of speech production," in *ICASSP '92*, pp. II13 – II16, IEEE, 1992.

[39] J. Schoentgen, "The spectral dynamics of a non-linear model of the glottal waveform," in *EUROSPEECH '89*, pp. 481–484, 1989.

[40] J. Schoentgen, "Non-linear signal representation and its application to the modelling of the glottal waveform," *Speech Communication*, vol. 9, pp. 189–201, June 1990.

[41] K. E. Cummings and M. A. Clements, "Analysis of glottal waveforms across stress styles," in *ICASSP '90*, pp. 369–372, IEEE, 1990.

[42] D. A. Cairns and J. H. L. Hansen, "Nonlinear analysis and classification of speech under stressed conditions," *Acoustical Society of America*, vol. 96, pp. 3392–3400, December 1994.

[43] C. Hamon, E. Moulines, and F. Charpentier, "A diphone synthesis system based on time-domain prosodic modifications of speech," in *ICASSP '89*, pp. 238–241, IEEE, 1989.

[44] A. Breen and J. Page, "Designing the next generation of text-to-speech systems," in *IEE Colloquium on Techniques for speech processing and their application*, no. 1994/138, pp. 6/1–6/5, IEE, June 1994.

[45] I. R. Titze, "Phonation threshold pressure: A missing link in glottal aerodynamics," *J Acoust Soc Am*, vol. 91, pp. 2926–2935, May 1992.

[46] J. Awrejcew, "Bifurcation portrait of human vocal oscillations," *Journal of Sound and Vibration*, vol. 136, no. 1, pp. 151–156, 1990.

[47] H. Herzel and J. Wendler, "Evidence of chaos in phonatory samples," in *EUROSPEECH 91*, vol. 1, pp. 263–266, 1991.

[48] I. Steinecke and H. Herzel, "Bifurcations in an asymmetric vocal-fold model," *Acoustical Society of America*, vol. 97, pp. 1874–1884, March 1995.

[49] J. Holzfuss and W. Lauterborn, "Lyapunov exponents from a time series of acoustic chaos," *Physical Review A*, vol. 39, pp. 2146–2152, February 1989.

[50] S. McLaughlin, S. Hovel, and A. Lowry, "Identification of nonlinearities in vowel generation," in *EUSIPCO 94*, vol. 2, pp. 1133–1137, Elsevier Science, 1994.

[51] J. W. A. Fackrell and S. McLaughlin, "The higher order statistics of speech signals," in *IEE Colloquium on Techniques for speech processing and their application*, no. 1994/138, pp. 7/1–7/6, IEE, June 1994.

[52] L. S. Liebovitch and T. Toth, "A fast algorithm to determine fractal dimension by box counting," *Physics Letters A*, vol. 141, pp. 386–390, November 1989.

[53] C. A. Pickover and A. Khorasani, "Fractal characterization of speech waveform graphs," *Comput and Graphics*, vol. 10, no. 1, pp. 51–61, 1986.

[54] L. Marcato and E. Mumolo, "Coding of speech signal by fractal techniques," in *EUROSPEECH '93*, pp. 745–748, 1993.

[55] M. F. Barnsley and A. D. Sloan, "A better way to compress images," *BYTE*, pp. 215–223, January 1989.

[56] E. L. J. Bohez, T. R. Senevirathne, and J. A. VanWinden, "Fractal dimension and iterated function system (ifs) for speech recognition," *Electronic Letters*, vol. 28, pp. 1382–1384, July 1992.

[57] P. A. Moakes and S. Beet, "Analysis of non-linear generating dynamics," in *ICSLP 94*, pp. 1039–1042, 1994.

[58] T. R. Senvirathne, E. L. J. Bohez, and J. A. VanWinden, "Amplitude scale method: New and efficient approach to measure fractal dimension of speech waveforms," *Electronics Letters*, vol. 28, pp. 420–422, February 1992.

[59] B. Townshend, "Nonlinear prediction of speech," in *IACSSP '91*, pp. 425–428, IEEE, 1991.

[60] T.Y.Li and J.Yorke, "Period three implies chaos," *American Mathematical Monthly*, vol. 82, pp. 985–92, 1975.

[61] D. Ruelle, "Strange attractors," *Mathematical Intelligencer*, vol. 2, pp. 126–37, 1980.

[62] F. Takens, *Dynamical Systems and Turbulence*, vol. 898 of *Lecture Notes in Mathematics*, pp. 366–381. Berlin: Springer, 1981.

[63] T. S. Parker and L. O. Chua, "Chaos: A tutorial for engineers," *Proceedings of the IEEE*, vol. 75, pp. 982–1008, August 1987.

[64] H. D. I. Abarbanel, "Chaotic signals and physical systems," in *ICASSP '92*, pp. IV113–IV116, IEEE, 1992.

[65] D. Kugiumtzis, B. Lillekjendlie, and N. Christophersen, "Chaotic time series part I: Estimation of invariant properties in state space," *Modelling, identification and control*, vol. 15, no. 4, 1994.

[66] L. R. Hunt and R. D. DeGroat, "Identifying nonlinear systems from experimatal data," in *ICASSP '91*, pp. 3517–3520, IEEE, 1991.

[67] J. P. Eckmann and D. Ruelle, "Ergodic theory of chaos and strange attractors," *Rev Mod Phys*, vol. 57, pp. 617–56, 1985.

[68] C. Myers, S. Kay, and M. Richard, "Signal separation for nonlinear dynamical systems," in *ICASSP '92*, pp. IV129–IV132, IEEE, 1992.

[69] J. S. Brush and J. B. Kazdtke, "Nonlinear signal processing using empirical global dynamical equations," in *ICASSP '92*, pp. V321 – V324, IEEE, 1992.

[70] J. I. Butterfield, "Fractal interpolation of radar signatures for detecting stationary targets in ground clutter," *IEEE AES Sytems Magazine*, vol. 6, pp. 10–14, July 1991.

[71] T. W. Frison, H. D. I. Abarbanel, J. Cembola, and R. Katz, "Nonlinear analysis of environmental distortions of continuous wave signals in the ocean," *Journal of the Acoustical Society of America*, vol. 99, pp. 139–146, January 1996.

[72] J. D. Farmer and J. J. Sidorowich, "Predicting chaotic time series," *Physical review letters*, vol. 59, no. 8, pp. 845–848, 1987.

[73] H. D. I. Abarbanel, R. Brown, and J. B. Kadtke, "Prediction and system identification in chaotic nonlinear systems: time series with broadband spectra," *Physics Letters A*, vol. 138, pp. 401–408, July 1989.

[74] H. D. I. Abarbanel, R. Brown, and J. B. Kadtke, "Prediction in chaotic nonlinear systems: Methods for time series with broadband fourier spectra," *Physical Review A*, vol. 41, pp. 1782–1807, February 1990.

[75] C. Myers, A. Singer, F. Shin, and E. Church, "Modeling chaotic systems with hidden markov models," in *ICASSP '92*, pp. IV565–IV568, IEEE, 1992.

[76] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase space reconstruction using the method of false nearest neighbours," *PHYSICA D*, 1992.

[77] T. Sauer, J. A. Yorke, and M. Casdagli, "Embedology," *Journal of Statistical Physics*, vol. 65, no. 3/4, pp. 579–616, 1991.

[78] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physical Review A*, vol. 33, pp. 1134–1140, February 1986.

[79] B. B. Mandelbrot, *The fractal geometry of nature*. W.H. Freeman and Company, 1977.

[80] J. D. Farmer, E. Ott, and J. A. Yorke, "The dimension of chaotic attractors," *PHYSICA D*, pp. 153–180, 1983.

[81] F. Hunt and F. Sullivan, "Efficient algorithms for computing fractal dimensions," in *Dimensions and entropies in chaotic systems*, pp. 74–81, Springer, 1986.

[82] G. Sugihara and R. May, "Nonlinear forecasting as a way of distinguishing chaos from measurement error in a data series," *Nature*, vol. 344, pp. 734–741, 1990.

[83] M. Casdagli, "Chaos and deterministic versus stochastic non-linear modelling," *Journal of the Royal Statistical Society B*, vol. 54, no. 2, pp. 303–328, 1991.

[84] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov exponents from a time series," *PHYSICA D*, vol. 16, pp. 285–317, 1985.

[85] D. S. Broomhead and G. P. King, *Nonlinear phenonema and chaos*, ch. On the qualitative analysis of experimental dynamical systems, pp. 113–144. Malvern science series, Adam Hilger, Bristol, 1986.

[86] M. Sano and Y. Sawada, "Measurement of the Lyapunov spectrum from a chaotic time series," *Physical review letters*, vol. 55, pp. 1082–1085, September 1985.

[87] H. D. I. Abardanel, R. Brown, and M. B. Kennel, "Local Lyapunov exponents computed from observed data," *Journal of Nonlinear Science*, vol. 1, pp. 175–199, 1991.

[88] T. S. Parker, "Insite: A software toolkit for studying chaos," in *Proceedings of the 34th Midwest symposium on circuits and systems*, pp. 752–755, IEEE, 1992.

[89] M. P. Kennedy, "Hardware toolkit for studying chaos," in *Proceedings of the 34th Midwest symposium on circuits and systems*, pp. 756–759, IEEE, 1992.

[90] J. P. Eckmann, S. O. Kamphorst, D. Ruelle, and S. Ciliberto, "Lyapunov exponents from time series," *Physical Review A*, vol. 34, pp. 4971–4979, December 1986.

[91] W. E. Weisel, "Continuous time algorithm for Lyapunov exponents i and ii," *Physical Review E*, vol. 47, pp. 3686–3697, May 1993.

[92] F. Rauf and H. M. Ashmed, "Calculation of Lyapunov exponents through nonlinear adaptive filters," in *IEEE International Symposium on circuits and systems*, pp. 568–571, IEEE, 1991.

[93] J. P. Crutchfield, J. D. Farmer, N. H. Packard, and R. S. Shaw, "Chaos," *Scientific American*, pp. 38–49, December 1986.

[94] M. J. Kearney and J. Stark, "An introduction to chaotic signal processing," *GEC Journal of Research*, vol. 10, no. 1, pp. 52–58, 1992.

[95] P. Grassberger and I. Procaccia, "Characterization of strange attractors," *Physical Reiview Letters*, vol. 50, pp. 346–349, January 1983.

[96] J. B. Ramsey and H. Yuan, "The statistical properties of dimension calculations using small data sets," *Nonlinearity*, vol. 3, pp. 155–176, 1990.

[97] J. Theiler, "Spurious dimension from correlation algorithms applied to limited time-series data," *Physical Review A*, vol. 34, pp. 2427–2432, September 1986.

[98] J. P. Eckmann and D. Ruelle, "Fundamental limitations for estimating dimensions and Lyapunov exponents in dynamical systems," *PHYSICA D*, vol. 56, pp. 185–187, 1992.

[99] J. Holzfuss and G. Mayer-Kress, "An approach to error-estimation in the application of dimension algorithms," in *Dimensions and entropies in chaotic systems*, pp. 114–122, Springer, 1986.

[100] R. L. Smith, "Estimating dimension in noisy chaotic time series," *Journal of the Royal Statistical Society B*, vol. 54, no. 2, 1992.

[101] G. Kember and A. C. Fowler, "Random sampling and the grassberger-procaccia algorithm," *Physics Letters A*, pp. 429–432, 1992.

[102] M. T. Rosenstein, J. J. Collins, and C. J. DeLuca, "A practical method for calculating the largest Lyapunov exponents from small data sets," *PHYSICA D*, vol. 65, pp. 117–134, 1993.

[103] D. S. Broomhead and G. P. King, "Extracting qualitative dynamics from experimental data," *PHYSICA D*, vol. 20, pp. 217–236, 1986.

[104] J. F. Gibson, J. Farmer, M. Casdagli, and S. Eubank, "An analytic approach to practical state space reconstruction," *PHYSICA D*, vol. 57, pp. 1–30, 1992.

[105] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge University Press, second ed., 1992.

[106] O. Michel and P. Flandrin, "An investigation of chaos-oriented dimensionality algorithms applied to ar(1) processes," in *ICASSP '92*, pp. V317 – V320, IEEE, 1992.

[107] M. Palus and I. Dvorak, "Singular-value decomposition in attractor reconstruction: pitfalls and precautions," *PHYSICA D*, vol. 55, pp. 221–234, 1992.

[108] R. Penrose, "A generalised inverse for matrices," in *Proc Camb Phil Soc 51*, pp. 406–413, 1955.

[109] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, 1985.

[110] E. N. Lorenz, "Deterministic non-periodic flow," *J. Atmos. Sci*, vol. 20, pp. 130–141, 1963.

[111] R. Brown, "Calculating Lyapunov exponents for short and/or noisy data sets," *Physical Review E*, vol. 47, pp. 3962–3969, June 1993.

[112] M.Banbrook, G.Ushaw, and S.McLaughlin, "Lyapunov exponents from a time series: a noise robust extraction algorithm." submitted to IEEE Trans Signal Processing, July 1995.

[113] M. Banbrook, G. Ushaw, and S. McLaughlin, "Lyapunov exponents from a time series: a noise-robust extraction algorithm," *Chaos, Solitons and Fractals*, 1996.

[114] M. Banbrook and S. McLaughlin, "Speech characterisation by nonlinear methods." submitted for rewiew to IEEE Trans on Speech and Audio Processing, 1996.

[115] M. Casdagli, "Nonlinear prediction of chaotic time series," *PHYSICA D*, vol. 35, pp. 335–356, 1989.

[116] G. Sugihara, "Nonlinear forecasting for the classification of natural time series," *Phil Trans of the Royal Society of London*, vol. 348, pp. 477–495, September 1994.

[117] H. M. Teager and S. M. Teager, "Evidence for nonlinear sound production mechanisms in the vocal tract," in *Proc NATO ASI on Speech Production and Speech Modelling*, pp. 241–261, 1990.

[118] A. M. Fraser, "Reconstructing attractors from scalar time series: a camparison of singular system and redundancy criteria," *PHYSICA D*, vol. 34, pp. 391–404, 1989.

[119] M. Banbrook and S. McLaughlin, "Is speech chaotic?: Invariant geometrical measures for speech data," in *IEE Colloquium on Exploiting Chaos in Signal Processing*, no. 1994/193, pp. 8/1–8/10, IEE, June 1994.

[120] J. H. Eggen, *On the quality of synthetic speech :evaluation and improvements*. PhD thesis, Technical University of Eindhoven, The Netherlands, September 1992.

[121] M. Banbrook and S. McLaughlin, "Speech characterisation by nonlinear methods," in *IEEE workshop on Nonlinear Signal and Image Processing*, pp. 396–400, 1995.

[122] E. Moulines and F. Charpentier, "Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Communication*, vol. 9, pp. 453–467, 1990.

[123] R. J. McAulay and T. H. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, pp. 744–754, August 1986.

[124] T. H. Quatieri and R. J. McAulay, "Speech transformations based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, pp. 1449–1464, December 1986.

[125] T. H. Quatieri and R. J. McAulay, "Shape invariant time-scale and pitch modification of speech," *IEEE Transactions on Signal Processing*, vol. 40, pp. 497–510, March 1992.

Appendix A
Analysis software

This appendix details the analysis software that was used within this Thesis all of which is supplied on the accompanying disk ('Software').

A.1 Short term prediction software

The code is stored as **casdagl.tgz** (sourcefiles tarred and gzipped).

The sourcecode named **'casd_pred.c'**, derived from the work by Casdagli [115], reports the short term prediction properties of a data set for a varying number of nearest neighbours. The following is a printout of the prompts supplied to the user when running the program.

```
************ cast_pred v3.2 9/11/94 *******
Please enter the input data filename :
Please enter the output data filename :
Is the data SVD reduced (y/n) :
Geometric or RMS mean or both (g or r or b) :
Enter the embedding dimension :
Enter the embedding m (delay) :
Enter the largest number of neighbours:
Enter the number of periods to average over  :
Enter the prediction period  :
Enter the number of data points to be used :
```

Data can be supplied as either a time series (set SVD reduced to 'n') or already embedded by setting 'SVD reduced' to 'y' which tells the program to expect data in `d` column format where `d` is the entered embedding dimension. Both files are expected to be ascii with rows of data seperated by newlines.

The other parameters are:

- *Geometric or RMS mean or both (g or r or b)* - The prediction error can be calculated as either the geometric mean or the root mean square error. In order to calculate the information dimension from the gradient this parameter must be set to 'g'.
- *Embedding dimension* - The number of dimensions into which the time series is embedded. Recommended that this be at least twice the expected dimension of the system under test.
- *Embedding delay (m)* - This is the delay used in the time delay embedding. This should be set using the first minima of the mutual information for the data (see 'Chaos_analyser).
- *Largest number of neighbours* - This is largest number of neighbours used to model the system (maximum of the x axis). This should be slightly smaller than the total number of points.
- *Periods to average over* - This is the number of different points around the attractor that are used as centers. The more points used the longer the program takes but the smoother the results should be.
- *Prediction period* - This is the number of steps ahead that the program predicts.
- *Number of points* - The numner of points contained in the data file.

The output is a two column ascii file, suitable for plotting (on a log/log scale) using GNUPLOT, which has number of neighbours .vs. prediction error.

A.2 Chaos analyser

The 'Chaos analyser' is a suite of programs bundled together under a common graphical user interface. The software supports the following features:

- Time series embedding
- 3 dimensional phase space viewer
- Animation of the attractor
- Mutual Information

- Singular Value Decompositon embedding
- Lyapunov exponents (FULL spectrum with noise robustness)
- Poincare sections
- Local singular value decomposition analysis

This software has been designed to work with a graphical interface (Xwindows) written in C on a Unix system (Suns).

The software can be found on the disk supplied ("software") under:

Chaos.tgz

and should be untarred using 'tar xvf Chaos.tar' after unzipping.

The format for the input data is a simple ascii file of the one dimensional time series.

The following are a number of screen shots showing a typical application.

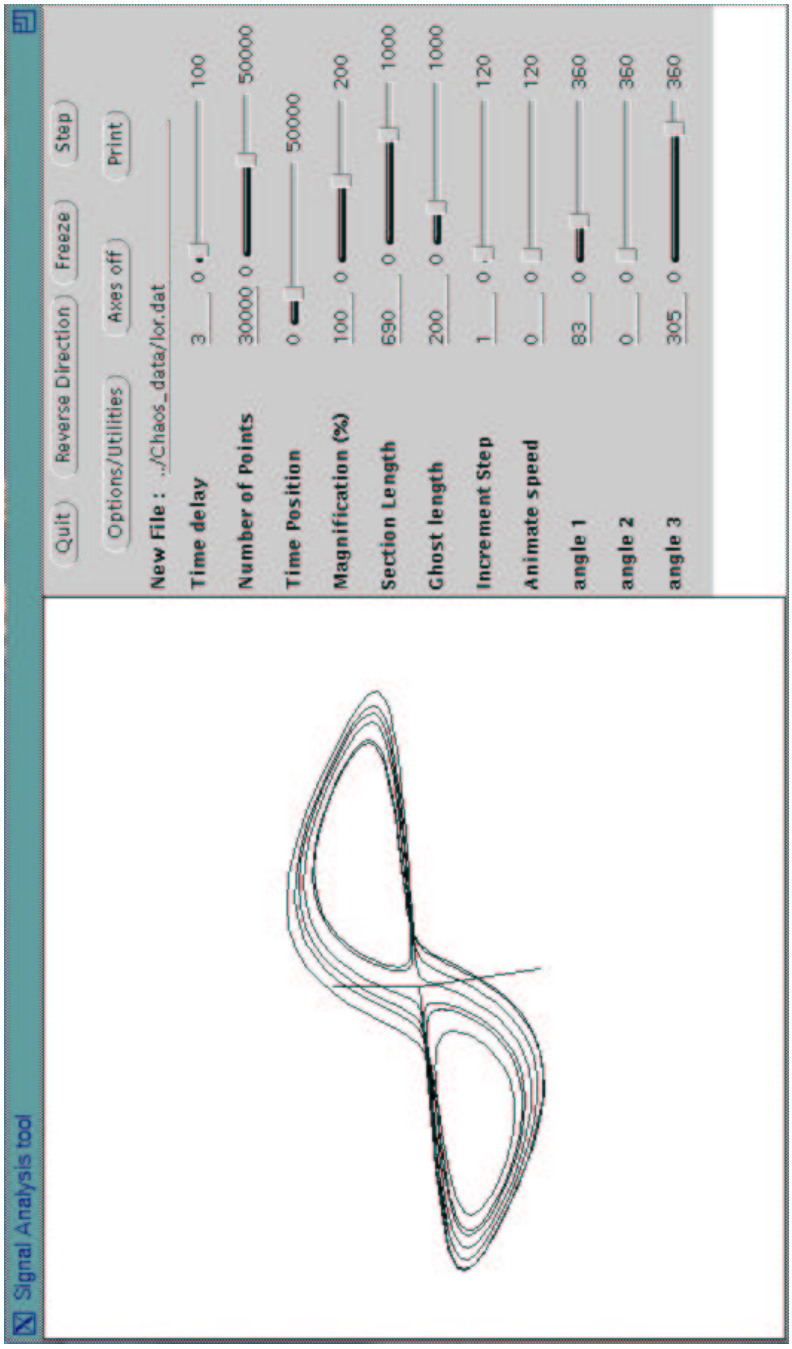


Figure A.1: *The main 3 dimensional phase space viewing window*

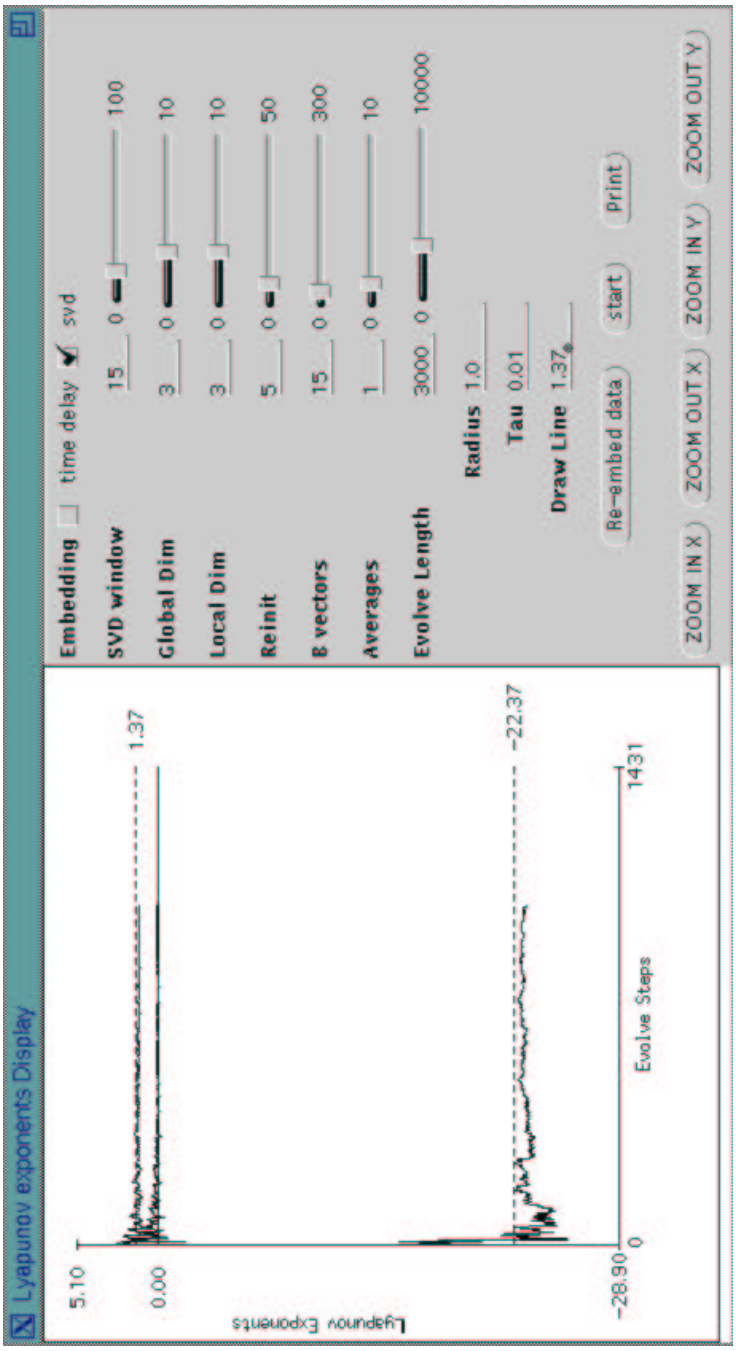


Figure A.2: *Lyapunov exponents for the Lorenz attractor*

Appendix B

The speech database

This appendix details the recording procedure and apparatus used for recording the speech database. The location of all the files refered to herein is in the Electrical Engineering file system of the University of Edinburgh under **/home/sspg**. In this directory there are a number of tools used for converting the files between different formats and the data itself which is arranged into subdirectories identified by the subjects initials. The file **/home/sspg/README.TXT** gives an overview of the current state of the directory and the tools within.

The recording equipment used consists of a 486, 33 MHz Elonex Personal Computer with an Ultrasound Max soundcard employing a sampling rate of 22 KHz. An Audio Technica ATM73a head mounted microphone was placed to the side of the subject's mouth to reduce wind noise. Special acquisition software [1] which allows for windowing of the input data enabled the capture of clean vowel sections with no co-articulation regions. To reduce possible noise contamination the subject is placed away from the computer in a sound reducing booth. The following document is the recording schedule given to the subject, detailing the procedure used and the phonemes requested.

[1] Phoneme acquisition software supplied by Alan Wrench of CSTR

Appendix C

Speech files

This appendix documents the synthesised speech that is contained on the disk supplied labelled 'Speech'.

The following files are all synthesised speech from the demonstration synthesiser described in Chapter 6:

- one.wav
- three.wav
- four.wav
- eight.wav
- eighteen.wav

The disk also contains an example of the BT Laureate synthesiser producing a sustained vowel sound (bt_18.wav).

Appendix D

Original publications

The work described in this thesis has been reported in the following publications (full reprints given in this appendix):

- M.Banbrook and S.McLaughlin, “Speech characterisation by nonlinear methods”, presented at IEEE workshop on Nonlinear Signal and Image Processing NSIP ’95, pp.396-400, June 1995.
- M Banbrook and S McLaughlin, "Is Speech Chaotic?: Invariant Geometric Measures for Speech Data", IEE Colloquium on "Exploiting Chaos in Signal Processing", Digest No 1994/193, pp8/1-8/10, June 1994
- M. Banbrook, G. Ushaw, S. McLaughlin," Lyapunov Exponents From A Time Series: A Noise-Robust Extraction Algorithm", to appear in Chaos, Solitons and Fractals, 1996.
- M. Banbrook and S. McLaughlin, “Dynamical modelling of vowel sounds as a synthesis tool”, to appear in ICSLP 96, 1996.

Papers submitted for review:

- M.Banbrook, G.Ushaw, S.McLaughlin, "How to calulate Lyapunov exponents from a short and noisy time series", submitted to IEEE trans on Signal Processing (July 1995).
- M.Banbrook, S.McLaughlin, "Speech Characterisation by Nonlinear Methods ", submitted to IEEE trans on Speech And Audio Processing (Jan 1996).